| | | |
|---|---|---|
| Application Serial No. | : | 10/788,590 |
| Filed | : | 27 February 2004 |
| Applicant | : | W. Voorhees et al. |
| Title | : | SYSTEMS AND METHODS FOR FLEXIBLE EXTENSION OF SAS EXPANDER PORTS |
| Art Unit | : | 2111 |
| Examiner | : | F. Zaman |
| Docket Number | : | 03-0605 |
| Date | : | 6 April 2007 |

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

## APPEAL BRIEF

Sir:

Appellants herewith file a Brief in support of their Appeal in the above identified matter. Also being submitted is the $500 fee under 37 CFR 41.20(b)(2) for the Appeal Brief.

## TABLE OF CONTENTS

## i. REAL PARTY IN INTEREST

The real party in interest is LSI Logic, Inc., the employer of the inventors at the time of the invention and the assignee of the patent rights in the above-identified matter.

## ii. RELATED APPEALS AND INTERFERENCES

No other appeals, interferences, or related applications are known to the Appellants, the Appellants' legal representative, or the Assignee, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

## iii. STATUS OF CLAIMS

Claims 1-9, 12-14, 17, and 18 stand rejected and remain in the application for consideration on appeal. The 35 U.S.C. §103(a) rejection of these claims forms the basis of this appeal.

## iv. STATUS OF AMENDMENTS

No amendments have been filed since the last office action (final office action) mailed 24 November 2006

## v. SUMMARY OF THE CLAIMED SUBJECT MATTER

A SAS ("serial attached SCSI") network typically comprises one or more SAS initiators coupled to one or more SAS targets via one or more SAS expander devices. In general, as is common in all SCSI communications, SAS initiators initiate communications with SAS targets. A target device may also re-connect to an initiator device that previously connected to it. The expander devices expand the number of ports of a SAS network domain used to interconnect SAS initiators and SAS targets (collectively referred to as SAS devices).

In designing a SAS expander device, it is common for a designer to create a design for a fixed number of expander ports. A variety of design points may be created such that a different number of ports are available in different product designs. However, changing a design to incorporate a greater or lesser number of ports may present numerous difficulties and associated costs to the designer. Increasing the number of ports in an existing design by a significant number may not be as simple as merely scaling parameters of a particular design.

The present invention provides methods and associated structures to permit flexible re-design of an expander to alter the number of ports in the expander design thus more easily generating a customized expander design. In one aspect, existing individual expander components are provided as standard circuit designs and coupled in a multi-chip module ("MCM") to a static internal fabric. Connections between ports of the multiple expander components in the MCM determine the static routes defined within the customized router. The number of ports in the MCM customized expander may be easily adapted by adding or removing expander components coupled to the internal fabric of the MCM. The internal fabric may then easily route one port to another within the MCM. Such an internal fabric may comprise fixed wires or other conductive paths through the MCM. The addition or subtraction of individual expander components within the MCM allows for simple re-design of an expander to accommodate virtually any desired number of ports. The internal fabric allows the customized ports to be configured with static routes in virtually any desired configuration.

More specifically, an exemplary embodiment of the invention of claim 1 provides for a multi-chip module (MCM 300 of FIG. 3) that includes a plurality of SAS expander component circuits (see FIG. 3 elements 306, 308, and 310 and page 6 third full paragraph). Each expander component of the MCM has a number (n) of internal ports internal to the MCM. Each expander component also a number (m) of external ports for coupling to SAS devices external to the MCM (see FIG. 3 ports 326 and 336 of component 306, ports 328 and 338 of component 308, and ports 330 and 340 of component 310; also see page 6 at line 5 of the third full paragraph). The MCM also includes an internal fabric (302 of FIG. 3) coupling together selected ones of the internal ports in selected ones of the plurality of SAS expander component circuits (see FIG. 5 coupling of internal ports 500..514 through fabric 592 and supporting text at page 9, last full paragraph through page 10, second full paragraph). The configuration of coupling together of the selected ones of the internal ports is static following initialization of the MCM (see page 7, second full paragraph through to the top of page 8). The MCM also includes coordination logic (see FIG. 3 logic 304) communicatively coupled to the plurality of SAS expander component circuits (e.g., via fabric 302) to coordinate operation of the plurality of SAS expander component circuits. The coordination logic is adapted to present a single expander to devices outside the module such that the single expander performs SCSI management protocol ("SMP") exchanges as a single SAS address (see page 8, second full paragraph).

Another exemplary embodiment provides a method for manufacturing a customized serial attached SCSI ("SAS") expander having a predetermined number of ports. The method as recited in claim 17 includes recitations that restrict the method to manufacturing of an MCM structure essentially as discussed above with regard to claim 1. The method includes disposing a number (N) of SAS expander components on a multi-chip module (MCM) (see element 402 of FIG. 4 and text at the first full paragraph of page 11). Each SAS expander component has a number (n) of internal ports internal to the MCM and wherein each SAS expander component has a number (m) of external ports for coupling to SAS devices external to the MCM (see above structural aspects discussed with regard to claim 1). The number of expanders disposed on the MCM (N) is sufficient to provide a total ports numbering ($m_N + n_N$) substantially equal to the predetermined number of ports (see element 400 of FIG. 4 and supporting text at the last full paragraph on page 10). The method then provides for disposing an internal fabric on the MCM (see element 404 of FIG. 4 and supporting text at the second full paragraph of page 11). The method then configures the internal fabric to provide desired routes between the total ports (see elements 406 through 414 of FIG. 4 and supporting text starting at the last full paragraph on page 11 through the first paragraph on page 12). Following the step of configuring, the routes between the total ports remains static at least until the MCM is reset (see above structural aspects discussed above with respect to claim 1). The method then provides for disposing a control logic circuit on the MCM coupled to the internal fabric (see element 408 of FIG. 4 and supporting text on page 12 first full paragraph). The control logic circuit performs SCSI management protocol ("SMP") exchanges as a single address for the customized SAS expander (see structural aspects discussed above with respect to claim 1). The step of configuring also includes applying signals from a control logic circuit to the internal fabric to configure the internal fabric as a static fabric at reset of the MCM (see structural limitations of the MCM circuit to be manufactured as discussed above with respect to claim 1).

## vi. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 1-9 and 12-14 are unpatentable under 35 U.S.C. § 103(a) over Bakke et al. (U.S. Patent Application Publication 2005/0071532) in view of Badamo et al. (United States Patent Publication 2002/0181476) and Seto (United States Patent Publication 2005/0138202).

2. Whether claims 17-18 are unpatentable under 35 U.S.C. § 103(a) over Bakke et al. (U.S. Patent Application Publication 2005/0071532) in view of Badamo et al. (United States Patent Publication 2002/0181476) and Seto (United States Patent Publication 2005/0138202).

## vii. ARGUMENT

### *Claims 1-9 and 12-14*

First and foremost, nothing in the art of record (considered individually or in any combination) teaches or reasonably suggests a SAS expander implemented as a Multi-Chip Module ("MCM") as claimed in claims 1-9 and 12-14. This discussion will focus on independent claim 1. None of the references describe an MCM as the term is understood by those of ordinary skill in the art. This term of art is well understood to describe a single integrated circuit that is manufactured to incorporate one or more other integrated circuit dies or components. Such MCM devices are generally fabricated using integrated circuit fabrication techniques - notably distinct from fabrication/assembly of printed circuit board cards and/or entire systems incorporating numerous such cards.

The references teach nothing of such an MCM structure for a SAS expander but rather teach structures of printed circuit cards and systems incorporating a plurality of such cards. In the Final Office Action, the Examiner maintains that this limitation is not found in the claims. The Examiner suggests in the Final Office Action that he is applying the "broadest reasonable interpretation" of the term "multi-chip module" to mean "any module containing multiple chips". Such an interpretation is ***clearly*** inconsistent with any reasonable understanding of the term of art by one of ordinary skill in the art. Under the Examiner's interpretation of a "multi-chip module", an entire personal computer is an MCM; a room-sized IBM System 360 of the 1960's is an MCM; the Space Shuttle is an MCM - clearly these are absurd examples of an MCM but within the Examiner's unreasonable interpretation and thus inconsistent with the normal, reasonable understanding of one of ordinary skill in the art. The Examiner's interpretation is ***UN***reasonably ***OVER***broad to a person of ordinary skill in the art.

Though there is no single, definitive source for definitions in this technological art, numerous glossaries provided by vendors and organizations in the field support the Applicants' narrower understanding of the term of art "multi-chip module" to mean an integrated circuit comprised of multiple other, interconnected chips.

One definition found in an on-line glossary provided by Dallas Semiconductors at www.maxim-ic.com/glossary defines the term of art as follows: "Multi-Chip Module: An integrated circuit package that contains two or more interconnected chips".

Another online glossary provided by Test Equipment Corporation at www.testeq.com/glossary/ defines "multi-chip module" as "A hybrid-type package containing a number of integrated circuits and other components. Used instead of printed circuit boards for applications calling for very high packing densities, high frequencies and high speeds of operation".

Another definition of the acronym MCM is provided by NASO Corporation at www.naso.com/pages/news_glossary.html as "MCM: Multi-Chip Module is a high density electronic assembly with interconnected silicon die".

Yet another definition provided by the Surface mount Technology Association (SEMTA) at www.smta.org/files/acronym_glossary.pdf states: "Multichip Module. A circuit comprised of two or more silicon devices bonded directly to a substrate by wire bond, TAB, of flip chip".

Still another definition is provided by the EDA Consortium (EDAC) at www.edac.org/industry_glossary.jsp#M as follows:

> MCM (Multi-Chip Module)
>
> A type of board technology whereby multiple, unpackaged integrated circuits (bare die) are mounted — along with signal conditioning or support circuitry such as capacitors and resistors — on a single laminate or ceramic base material. The MCM footprint is much smaller than conventional single chip packages, resulting in a smaller motherboard and smaller space requirements for panels, enclosures, and cabling. The result is a high-density module that resembles a single component when mounted on a printed circuit board. By combining more circuit functions in an MCM, fewer system assemblies are required, resulting in lower circuit design costs, integrated functional testing, and higher manufacturing yields.

These and numerous other similar definitions evidence the well understood meaning of the term of art "multi-chip module" (MCM) as an integrated circuit or circuit package containing multiple interconnected circuit dies. A printed circuit board such as cited in Bakke by the Examiner is NOT understood to be an MCM by those of ordinary skill in the art - let alone a system comprising multiple such printed circuit boards. Thus nothing in Bakke or any of the art of record, considered individually or in any combination, teaches or reasonably suggests the claimed structure of claim 1 - namely an MCM comprising the recited elements.

Applicants recognize that the mere integration of previously distinct, discrete electronic components is not, without more, a patentable distinction. In the invention of claim 1, for example, various benefits are derived from this MCM architecture SAS expander - more than the mere integration of previously distinct/discrete components. Specific benefits attributable to the recited MCM integrated architecture that are novel and non-obvious are discussed in the specification of the subject application and are also recited in the claims (including for example claim 1). These benefits are either dismissed by the Examiner or improperly read on the prior art.

Bakke individually or in the proposed combination with Badamo fails to disclose coordination logic as recited in claim 1 that provides static routing features as claimed. The Examiner points to Badamo paragraph 0041 in support of his assertion that the "routing" of FC(s) 20 in Badamo's figure 3 teaches such a static routing feature. Quite the opposite is taught by Badamo. Paragraph 0041 of Badamo in reference to his figure 3 reads as follows:

> Any line card 22 can send traffic to any service card 24. This routing can be configured statically or can be determined dynamically by the line card 22. Any service card 24 can send traffic requiring ingress processing (e.g. from SC1 24' to SC2 24") to any other service card 24 for ingress processing. Line cards 22 with the capability to classify ingress traffic can thus make use of unused capacity on the ingress service cards 24 by changing the routing.

Though the word "static" is used by Badamo, it is clear to one of ordinary skill in the art that the "routing" performed by FC 20 of Badamo is "dynamic" or "flexible" ("Any line card 22 can send traffic to any service card 24"). This is the very essence of a dynamic routing capability rather than static coupling as recited in claim 1.

Badamo's paragraph 0039 just above paragraph 0041 further supports the dynamic routing capability of Badamo's FC 20 component stating (emphasis added): "The ***flexible routing*** therefore enables any service card 24 or line card 22, in particular a spare service card 24 or line card 22, to assume the role of another service card 24 or line card 22 ***by only changing the routing through the switch fabric card (FC) 20***". Again, this is the very essence of dynamic routing as understood by those of ordinary skill in the art. Such dynamic routing is a fundamental purpose of a typical, stand alone SAS expander (or essentially any stand alone network switch appliance).

By contrast, rejected claim 1 (for example) clearly recites that the internal fabric of the MCM that couples selected internal ports of the plurality of SAS expander components "is static following initialization of the MCM". Since the plurality of SAS expander components are coupled internally to the MCM and present themselves as a single, integrated SAS expander (as discussed further below), the routes between the various expander components of the MCM are fixed or static following initialization of the MCM - there is no dynamic or flexible routing as required for the structures and methods of Badamo and Bakke. It is unreasonable for the Examiner to read the routing features of Badamo's FC 20 that are clearly dynamic and flexible in nature as teaching the recited static routing of, for example, rejected claim 1. This feature/benefit is derived from the MCM structure of claim 1 and is not generally available from a system such as Badamo or Bakke comprising a plurality of system components as complete, discrete printed circuit boards. Thus this benefit is neither taught nor reasonably suggested by Bakke, Badamo, or any of the art of record, either considered individually or in any combination.

The prior art of record also fails to show a benefit of the claimed structure wherein the plurality of SAS expander components (coupled internally by a statically routed fabric) present to external devices a single integrated SAS expander as in SMP protocol exchanges. In SCSI Management Protocol ("SMP") a device responds to a single SAS address for management related exchanges. A stand-alone, discrete, SAS expander (not integrated as claimed into an MCM structure) would normally respond to its own unique SAS address. However, the invention of rejected claim 1 recites that the plurality of SAS expander components as configured in an MCM with the recited coordination logic forces the plurality of SAS expander components to present the plurality of expander components as a single, integrated SAS expander - i.e., a plurality of SAS expanders that interact and respond to a single SAS address for SMP exchanges - not multiple independent SAS addresses. Thus, to a management application, the MCM appears as a single integrated SAS expander although it comprises (as recited) multiple SAS expander components.

In the new grounds for rejection of this Final Office Action, the Examiner adds Seto to the combination of Bakke and Badamo and urges that Seto teaches the feature of presenting such a single integrated SAS expander for SMP exchanges. Apparently the Examiner fails to understand that the claimed MCM comprises multiple SAS expander components. As noted, such multiple SAS expanders (like any SAS devices) each respond to its own SAS address as regards SMP exchanges. Only with the recited coordination logic do the multiple SAS expander components of rejected claim 1 present themselves collectively as a single SAS expander with a single SAS address in SMP exchanges. Seto teaches nothing more in this regard than an example of ***any*** well known SAS device that includes a capability to perform SMP exchanges - i.e., any stand-alone, discrete, standard SAS device. The Examiner points to element 38C of Seto's figure 2 and element 180 of figure 5A and associated text at paragraphs 0014 and 0023, respectively in support of his reading. The cited portions of Seto do not show a device that has multiple SAS expanders (or other SAS devices) that respond as a single SAS device with a single SAS address in SMP exchanges. Seto's figure 2 and associated text in paragraph 0014 shows an "adaptor" [sic] 12 that includes multiple adapters (12a and 12b of figure 1). Each of the adapters 12a and 12b may include an SMP link layer 38c. Nothing in Seto suggests that the two adapters 12a and 12b share a common SAS address in their SMP exchanges such that external device would perceive a single integrated SAS expander (or other SAS device). Rather, adapter 12a and 12b would each provide its own SMP layer 38c (and 40c and 48c) and its own corresponding SAS address in its respective SMP exchanges. Adapter 12 of Seto is a simple physical packaging of multiple SAS devices on a single circuit board or system structure. Nothing in Seto suggests that this mere physical integration (on a board or system) of multiple adapters 12a and 12b includes coordination logic as claimed in rejected claim 1 to present two adapters (e.g., 12a and 12b) as a single integrated adapter with a single SAS address in SMP exchanges.

Seto's figure 5A and associated text at paragraph 0023 merely discusses the standard SAS architecture in which each SAS device (180, 182, 184, and 186 of figure 5A) has a unique SAS address ("x", "A", "B", and "C", respectively). As is well known, a single SAS device may comprise multiple PHYs (e.g., every SAS expander includes multiple PHYs). Each PHY represents a physical connection that may be used to communicate with another SAS device. A logical SAS port may comprise one or more such PHYs of the SAS device. Each such port may have its own unique SAS address for purposes of SAS data exchanges. However, such a standard SAS device, regardless of the number of PHYs therein, responds to SMP exchanges only using the single unique address of the SAS device as a whole. Again, as above, the claimed MCM includes a plurality of SAS expanders (each of which has multiple PHYs or ports as is axiomatic for a SAS expander). Each of these plurality of SAS expanders in the claimed MCM would normally have a unique SAS address for SMP exchanges (just as devices 180 through 186 of Seto's figure 5A each has a unique SAS address may respond using its unique SAS address). However, the claimed invention further provides the recited coordination logic such that, for purposes of SMP exchanges, the recited multiple SAS expanders interact as a single integrated SAS expander. This feature is neither taught nor reasonably suggested by Bakke, Badamo, Seto, or any of the art of record, either considered individually or in any combination.

In view of the above discussion, Applicants maintain that independent claim 1 is allowable over the combination of Bakke, Badamo, and Seto, and all art of record, considered individually or in any combination. Dependent claims 2-9 and 12-14 recite additional novel features. As regards the additional features of claims 6-9, the Examiner adds the teachings of Barrow to the prior combination of Bakke, Badamo, and Seto to suggest these additional features are not patentable. For at least the same reasons as discussed above for claim 1, dependent claims 2-9 and 12-14 (dependent from base claim 1) are allowable over all art of record and allowable as dependent from allowable base claim 1.

***Claims 17-18***

Claims 17 and 18 recite methods for manufacturing a customized SAS expander. The method of claim 17, for example, includes steps to dispose a number (N) of SAS expander components on an MCM, disposing an internal fabric on the MCM, disposing control logic on the MCM and configuring the internal fabric disposed on the MCM. Claim 18 recites a related method. Thus the methods recite steps to produce a customized SAS expander as an MCM such as the MCM discussed above with regard to claims 1-9 and 12-14. The MCM resulting from the manufacturing steps is recited as including structural and functional limitations corresponding substantially to the structures and functions discussed above with regard to claims 1-9 and 12-14.

The Examiner applies the same combination of Bakke, Badamo, and Seto to reject the methods of claims 17-18 in essentially the same manner as they are applied to the rejection of the structures of claims 1-9 and 12-14. Since the teachings of Bakke, Badamo, and Seto fail to teach or reasonably suggest the MCM structure that results from the claims method steps of claims 17 and 18, Applicants respectfully submit that the combination fails to teach the recited method. Thus Applicants submit claims 17 and 18 are allowable over all art of record, considered individually or in any combination.

## viii. CLAIMS APPENDIX

1. A multi-chip module (MCM) comprising:

a plurality of serial attached SCSI ("SAS") expander component circuits each having a number (n) of internal ports internal to the MCM and each having a number (m) of external ports for coupling to SAS devices external to the MCM;

an internal fabric coupling together selected ones of the internal ports in selected ones of the plurality of SAS expander component circuits wherein the configuration of coupling together of the selected ones of the internal ports is static following initialization of the MCM; and

coordination logic communicatively coupled to the plurality of SAS expander component circuits to coordinate operation of the plurality of SAS expander component circuits wherein the coordination logic is adapted to present a single expander to devices outside the module, wherein the single expander performs SCSI management protocol ("SMP") exchanges as a single SAS address.

2. The module of claim 1 wherein the plurality of SAS expander component circuits comprises a number ($N$) of SAS expander components each having a number ($n_N$) of internal ports.

3. The module of claim 1 wherein the plurality of SAS expander component circuits comprises a number ($N$) of SAS expander components each having a number ($m_N$) of external ports.

4. The module of claim 1 wherein the internal fabric comprises a static fabric.

5. The module of claim 4 wherein the static fabric is configured at manufacture of the MCM.

6. The module of claim 1 wherein the internal fabric is initially configured at reset of the MCM.

7. The module of claim 6 further comprising:

a control logic circuit to configure the internal fabric at reset of the MCM.

8. The module of claim 1 wherein the internal fabric comprises a programmable fabric.

9. The module of claim 8 wherein the programmable fabric is adapted to be configured by information received from a SAS device coupled to an external port of a SAS expander of the MCM.

(Claims 10 and 11 are canceled)

12. The module of claim 1 wherein the coordination logic is adapted to coordinate SCSI management protocol ("SMP") message processing logic within each expander of the plurality of SAS expander component circuits.

13. The module of claim 1 wherein the coordination logic is adapted to present a single SAS address for the plurality of SAS expander component circuits.

14. The module of claim 1 wherein the coordination logic is adapted to present a single set of PHY numbers for the PHYs of the plurality of SAS expander component circuits.

(Claims 15 and 16 are canceled)

17. A method for manufacturing a customized serial attached SCSI ("SAS") expander having a predetermined number of ports, the method comprising:

disposing a number (N) of SAS expander components on a multi-chip module (MCM) wherein each SAS expander component has a number (n) of internal ports internal to the MCM and wherein each SAS expander component has a number (m) of external ports for coupling to SAS devices external to the MCM and wherein the number N is sufficient to provide a total ports numbering ($m_N + n_N$) substantially equal to the predetermined number of ports;

disposing an internal fabric on the MCM;

configuring the internal fabric to provide desired routes between the total ports wherein following the step of configuring, the routes between the total ports remains static at least until the MCM is reset; and

disposing a control logic circuit on the MCM coupled to the internal fabric, wherein the control logic circuit performs SCSI management protocol ("SMP") exchanges as a single address for the customized SAS expander,

wherein the step of configuring further comprises:

applying signals from a control logic circuit to the internal fabric to configure the internal fabric as a static fabric at reset of the MCM.

18. A method for manufacturing a customized serial attached SCSI ("SAS") expander having a predetermined number of ports, the method comprising:

disposing a number (N) of SAS expander components on a multi-chip module (MCM) wherein each SAS expander component has a number (n) of internal ports internal to the MCM and wherein each SAS expander component has a number (m) of external ports for coupling to SAS devices external to the MCM and wherein the number N is sufficient to provide a total ports numbering ($m_N + n_N$) substantially equal to the predetermined number of ports;

disposing an internal fabric on the MCM;

configuring the internal fabric to provide desired routes between the total ports wherein following the step of configuring, the routes between the total ports remains static at least until the MCM is reset; and

disposing a coordination logic circuit on the MCM communicatively coupled to the SAS expander components to coordinate operation of the plurality of SAS expander components to present a single expander interface to devices external to the MCM, wherein the coordination logic circuit performs SCSI management protocol ("SMP") exchanges as a single address for the customized SAS expander.

## xi. EVIDENCE APPENDIX

Included are copies of the evidence relied upon by the Examiner as to the grounds of the rejections under 35 U.S.C. §103(a) to be reviewed on appeal.

1. Bakke et al. (United States Patent Application Publication 2005/0071532).

2. Badamo et al. (United States Patent Application Publication 2002/0181476).

3. Seto (United States Patent Application Publication 2005/0138202).

4. Barrow et al. (United States Patent Application Publication 2002/0188786).

## x. RELATED PROCEEDINGS APPENDIX

None.

## SUMMARY

Appellants argue that the Examiner's rejections of claims 1-9, 12-14, and 17-18 under 35 U.S.C. §103(a) are inadequate as a matter of law and should be reversed. It is believed that this Appeal Brief has been timely filed within two (2) months of receipt of the Notice of Appeal (electronically filed 14 February 2007). However, if an extension of time is deemed to be required by the Patent Office, the Patent Office is hereby requested to accept this request as a petition for a one (1) month extension of time to respond with any requisite fees therefore being charged to deposit account 12-2252.

Respectfully submitted,

/Daniel N. Fishman/

Daniel N. Fishman #35,512
Duft, Bornsen & Fishman, LLP
1526 Spruce St.
Suite 302
Boulder, CO 80302
(303) 786-7687
(303) 786-7691 (fax)

US 20050071532A1

(19) **United States**
(12) **Patent Application Publication**
**Bakke et al.**
(10) **Pub. No.: US 2005/0071532 A1**
(43) **Pub. Date: Mar. 31, 2005**

(54) **METHOD AND APPARATUS FOR IMPLEMENTING RESILIENT CONNECTIVITY IN A SERIAL ATTACHED SCSI (SAS) DOMAIN**

(75) Inventors: **Brian Eric Bakke**, Rochester, MN (US); **Timothy Jerry Schimke**, Stewartville, MN (US)

Correspondence Address:
**IBM CORPORATION**
**ROCHESTER IP LAW DEPT. 917**
**3605 HIGHWAY 52 NORTH**
**ROCHESTER, MN 55901-7829 (US)**

(73) Assignee: **INTERNATIONAL BUSINESS MACHINES CORPORATION**, ARMONK, NY (US)

(21) Appl. No.: **10/670,710**

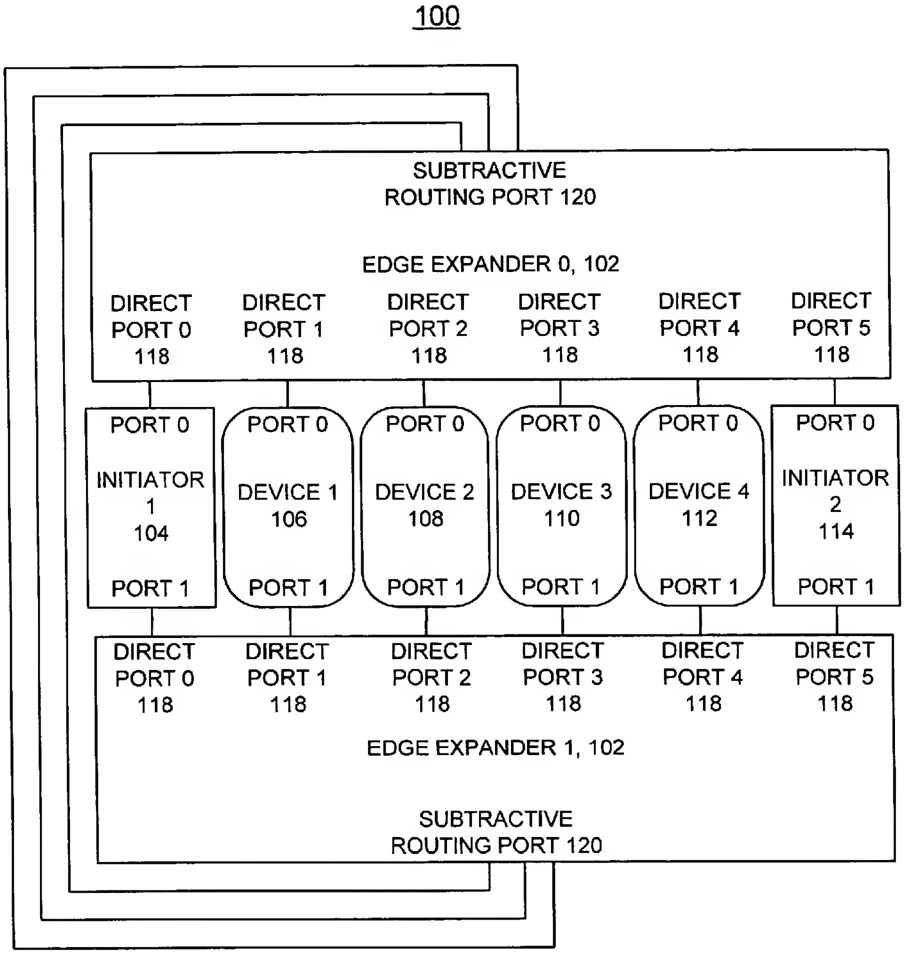
(22) Filed: **Sep. 25, 2003**

**Publication Classification**

(51) **Int. Cl.$^{7}$** .......... **G06F 13/00**
(52) **U.S. Cl.** .......... **710/300**

(57) **ABSTRACT**

A method and apparatus are provided for implementing resilient connectivity in a Serial Attached SCSI (SAS) domain. A first edge expander is connected to a first port of a plurality of SAS devices for enabling communications between each of the plurality of SAS devices through the first edge expander. A second edge expander is connected to a second port of the plurality of SAS devices for enabling communications between each of the plurality of SAS devices through the second edge expander. A subtractive routing port of each of the first edge expander and the second edge expander is connected together for enabling communications between each of the plurality of SAS devices via the first ports and the second ports of the plurality of SAS devices.



PRIOR ART

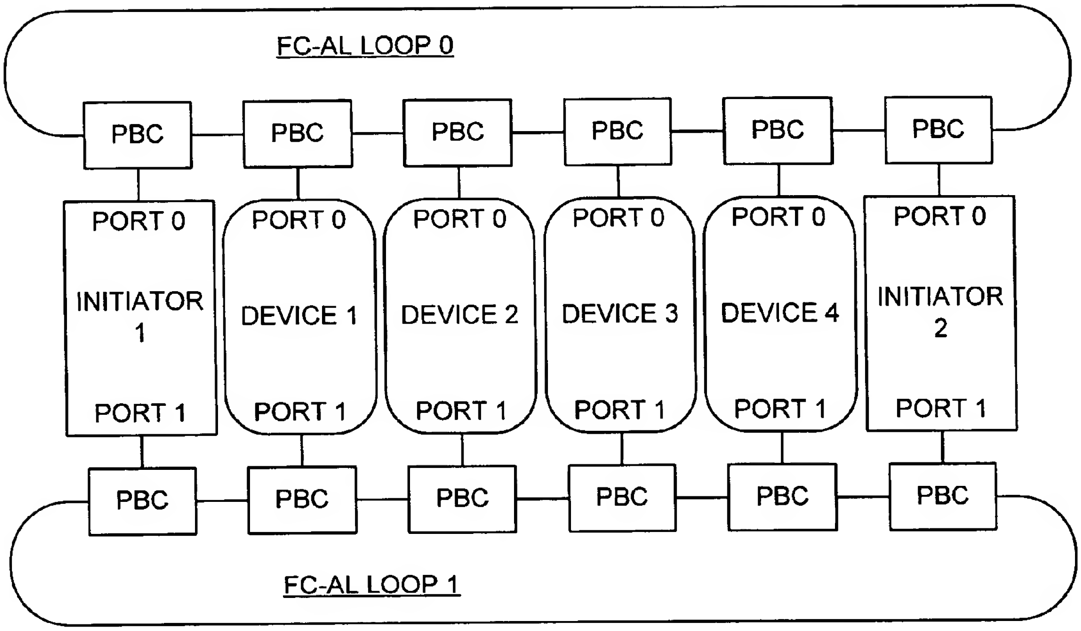


FIG. 1

PRIOR ART

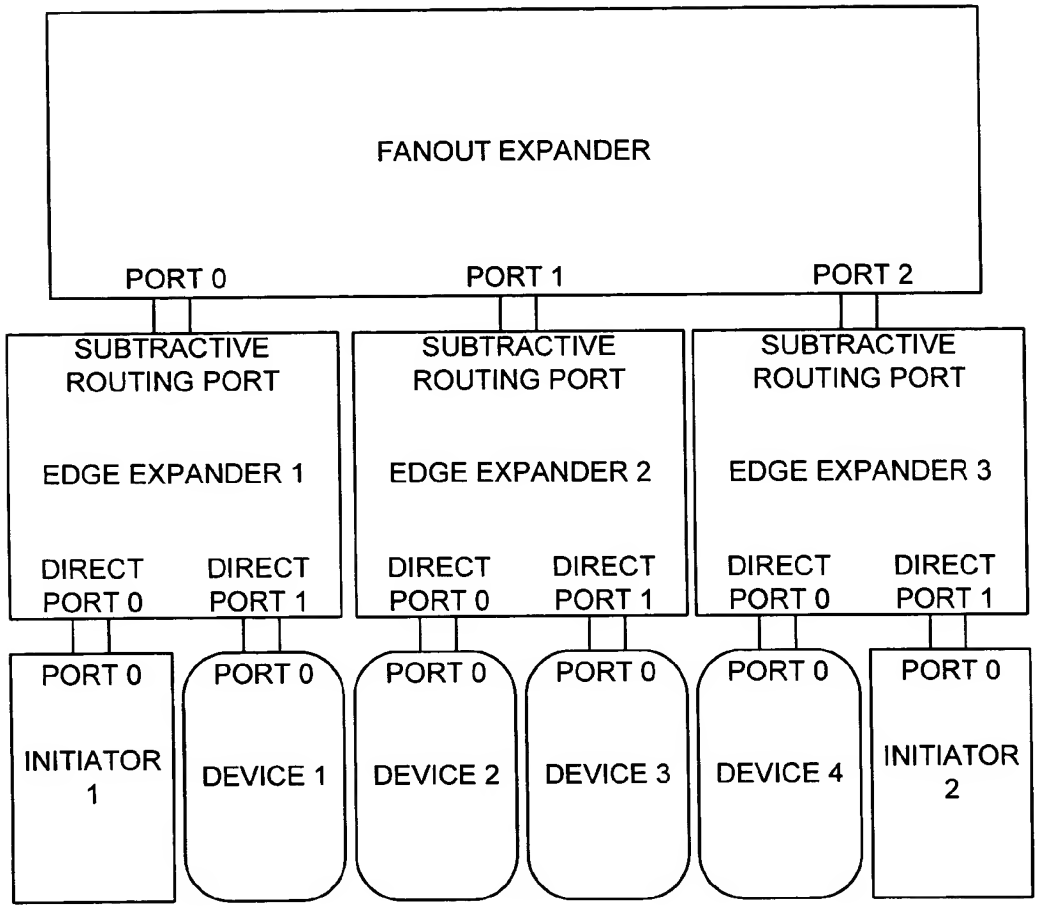


FIG. 2

PRIOR ART

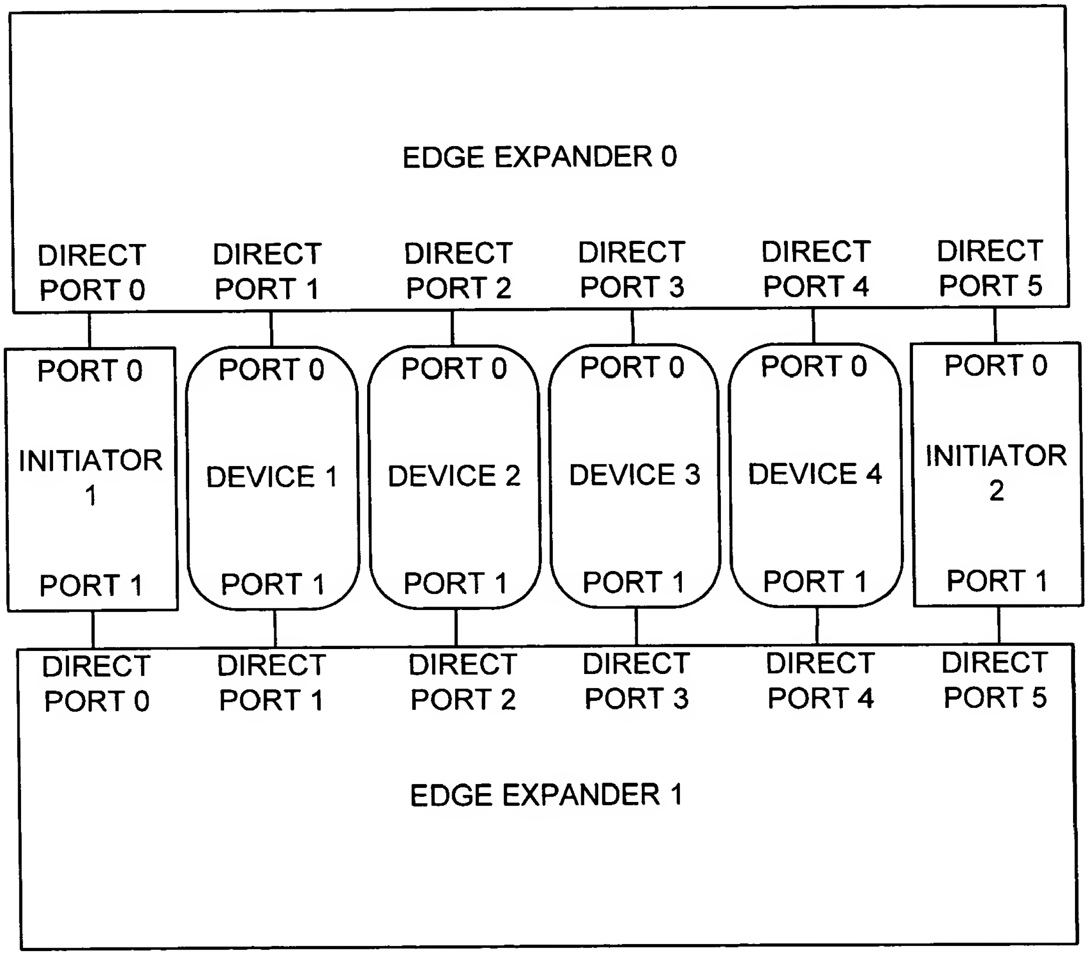


FIG. 3

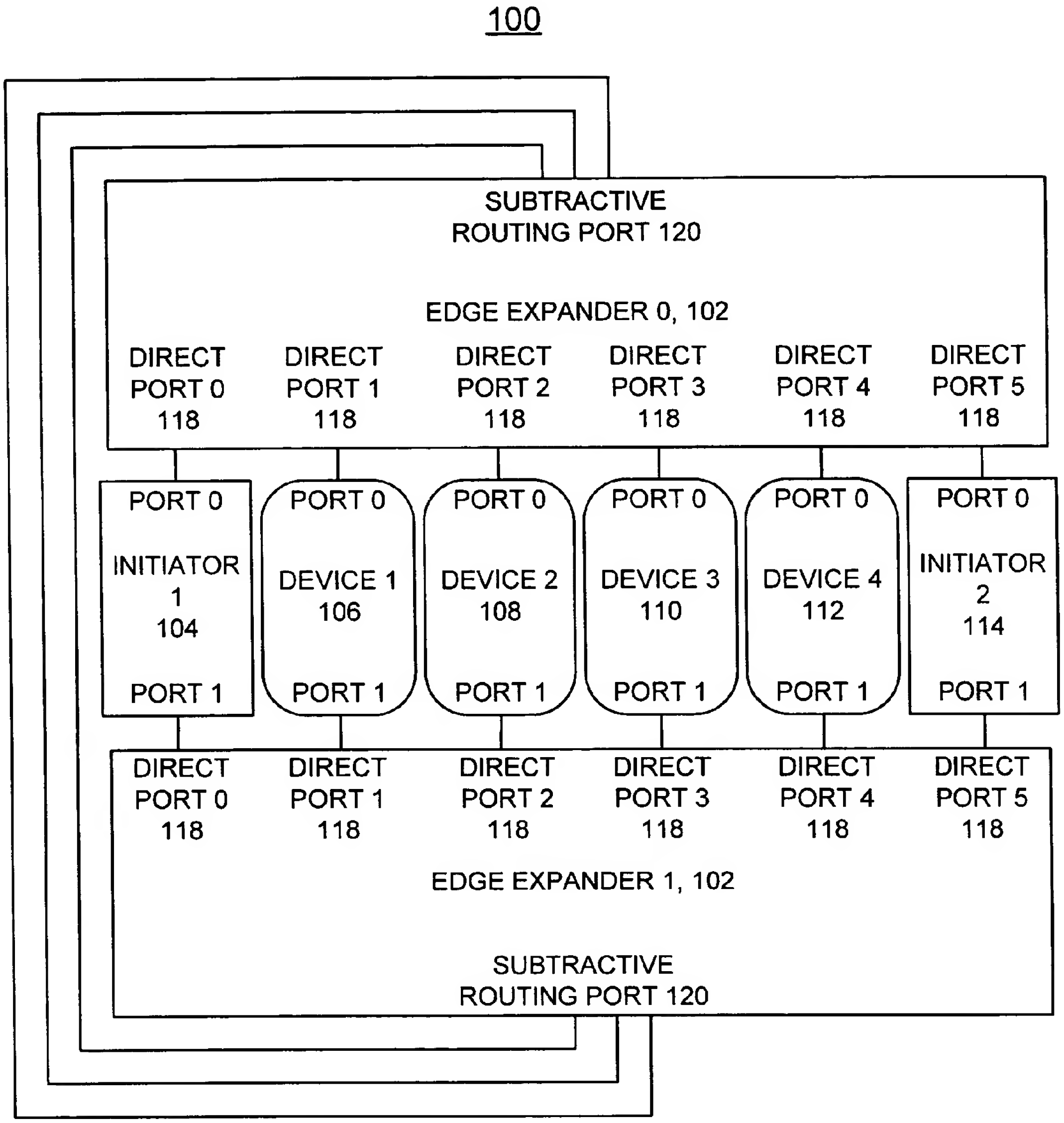

100
SUBTRACTIVE ROUTING PORT 120
EDGE EXPANDER 0, 102
DIRECT PORT 0 118
DIRECT PORT 1 118
DIRECT PORT 2 118
DIRECT PORT 3 118
DIRECT PORT 4 118
DIRECT PORT 5 118
PORT 0
INITIATOR 1 104
PORT 1
PORT 0
DEVICE 1 106
PORT 1
PORT 0
DEVICE 2 108
PORT 1
PORT 0
DEVICE 3 110
PORT 1
PORT 0
DEVICE 4 112
PORT 1
PORT 0
INITIATOR 2 114
PORT 1
DIRECT PORT 0 118
DIRECT PORT 1 118
DIRECT PORT 2 118
DIRECT PORT 3 118
DIRECT PORT 4 118
DIRECT PORT 5 118
EDGE EXPANDER 1, 102
SUBTRACTIVE ROUTING PORT 120

FIG. 4

# METHOD AND APPARATUS FOR IMPLEMENTING RESILIENT CONNECTIVITY IN A SERIAL ATTACHED SCSI (SAS) DOMAIN

## FIELD OF THE INVENTION

[0001] The present invention relates generally to the data processing field, and more particularly, relates to a method and apparatus for implementing resilient connectivity in a Serial Attached SCSI (SAS) domain.

## Description of the Related Art

[0002] The problem to be solved is to maintain availability of access between storage nodes, with a node being either an initiator device, for example, to connect to a host system, or target device, such as, a disk or tape drive. Loss of access is a problem because it causes loss of access to the data contained on the media, and is also a problem in configurations with multiple initiator adapters because an initiator adapter may not be able to detect the presence of another initiator adapter and may make erroneous decisions as a result.

[0003] The best existing solutions are generally tolerant of a single failure, however access is lost if more than one failure is encountered. **FIGS. 1, 2**, and **3** illustrate the drawbacks of the current technologies that allow for redundant paths.

[0004] **FIG. 1** illustrates a solution based upon Fiber Channel Arbitrated Loop (FC-AL) devices. The FC-AL devices are connected in loop topologies, and are dual-ported for reliability reasons utilizing two FC-AL loops **0**, **1**. A port bypass circuit (PBC) is used to maintain loop connectivity in the presence of missing or failed nodes on the loop. A failure in any single component is tolerated with access being maintained. However, the existence of multiple failures such as an initiator port failure on one loop and a target port failure on the other loop, for example, failures of initiator **1** Port **0** and Device **1** Port **1**, will cause the initiator and target devices to be unable to communicate.

[0005] Serial Attached SCSI (SAS) is an emerging industry standard that is targeted to replace parallel SCSI devices as the enterprise-class standard storage interface. An interconnection of Serial Attached SCSI (SAS) nodes is known as a SAS domain. The SAS devices are interconnected with a set of point-to-point links in the SAS domain. SAS devices also have two connections for performance and reliability reasons.

[0006] **FIG. 2** illustrates a configuration based upon Serial Attached SCSI (SAS) devices. The illustrated configuration contains one SAS domain with the two ports of each SAS device attached to a single edge expander, as might be used in a low cost system. The edge expander is a device that allows fanout and connections between multiple devices. In this configuration a failure of a single component, such as an edge expander, causes access to be lost to the media device.

[0007] As shown in **FIG. 2**, in an interconnection of Serial Attached SCSI (SAS) nodes or SAS domain, each logical connection to a node is made via a port. A port is composed of point-to-point links, which are denoted as phys. For performance and reliability reasons multiply phys may be ganged together to make up a port, this allows for multiple concurrent connections to be established. A typical media device, such as a disk drive, is expected to contain two ports with each port composed typically of a single phy.

[0008] To enable larger configurations, edge expanders are used. The edge expander enables communication to be established between nodes that are directly connected to the edge expander. The edge expander is a simple device with significantly less function than a switch would have, for example, the edge expander has no routing tables, and the edge expander is available at a significantly lower cost as a result. For example, the projected cost estimates for an edge expander are less than **10**% of a fiber channel switch cost per port.

[0009] To enable larger configurations than would be allowed by direct connections to the edge expander, that is to construct SAS domains with a greater number of nodes, each edge expander contains a subtractive routing port as shown in **FIG. 2**. If a SAS node makes a connection request to the edge expander requesting a node that is not directly connected to the edge expander, the request is then forwarded out this subtractive routing port. The respective subtractive routing ports of the edge expanders are connected to a fanout expander. The fanout expander does contain a routing table, and is able to determine the correct edge expander to route the request, that is, the edge expander to which the requested node is directly connected.

[0010] **FIG. 3** illustrates a more advanced SAS configuration utilizing an interconnect strategy similar to that used for FC-AL systems. There is now no longer a single point of failure since an edge expander failure now only impacts one of the two connections to each SAS device. However, as was seen in the FC-AL case, a pair of failures such as an initiator port failure coupled with a target port failure causes a loss of all communication between the initiator and target, for example failures of initiator **1** Port **0** and Device **1** Port **1**.

[0011] A need exists for a mechanism for implementing resilient connectivity in a Serial Attached SCSI (SAS) domain.

## SUMMARY OF THE INVENTION

[0012] A principal object of the present invention is to provide a method and apparatus for implementing resilient connectivity in a Serial Attached SCSI (SAS) domain. Other important objects of the present invention are to provide such method and apparatus for implementing resilient connectivity in a Serial Attached SCSI (SAS) domain substantially without negative effect and that overcome many of the disadvantages of prior art arrangements.

[0013] In brief, a method and apparatus are provided for implementing resilient connectivity in a Serial Attached SCSI (SAS) domain. A first edge expander is connected to a first port of a plurality of SAS devices for enabling communications between each of the plurality of SAS devices through the first edge expander. A second edge expander is connected to a second port of the plurality of SAS devices for enabling communications between each of the plurality of SAS devices through the second edge expander. A subtractive routing port of each of the first edge expander and the second edge expander is connected together for enabling communications between each of the plurality of SAS devices via the first ports and the second ports of the plurality of SAS devices.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The present invention together with the above and other objects and advantages may best be understood from the following detailed description of the preferred embodiments of the invention illustrated in the drawings, wherein:

[0015] **FIGS. 1, 2,** and **3** illustrate prior art technologies that allow for redundant paths; and

[0016] **FIG. 4** is a block diagram illustrating a Serial Attached SCSI (SAS) configuration or system including a resilient SAS domain in accordance with the preferred embodiment.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0017] Referring now to the drawings, in **FIG. 4** there is shown a Serial Attached SCSI (SAS) network generally designated by the reference character **100** including a resilient SAS domain in accordance with the preferred embodiment. In the SAS network **100**, all connections are made via point-to-point links. SAS network **100** includes a pair of edge expanders **0, 1, 102** used to interconnect a plurality of SAS initiators **1-2**, and devices **1-4, 104, 106, 108, 110, 112,** and **114**.

[0018] Each of the edge expanders **0, 1, 102** contains multiple ports including a plurality of direct routing ports, **0-5, 118** and a subtractive routing port **120**. Each node of the SAS domain including respective ports **0** and **1** of initiator **1, 104**, devices **1-4, 106, 108, 110, 112,** and initiator **2, 114** is connected to a respective edge expander direct routing port **118** of the respective edge expanders **0, 1, 102**. The respective direct routing ports **0-5, 118** of the edge expanders **0, 1, 102** are respectively connected via point-to-point links to respective port **0**, port **1** of SAS initiator **1, 104**, SAS device **1, 106**, SAS device **2, 108**, SAS device **3, 110**, SAS device **4, 112**, and SAS initiator **2, 114**. Communications between the ports **0** of initiator **1, 104**, devices **1-4, 106, 108, 110, 112,** and initiator **2, 114** are provided through the edge expander **0, 102**. Communications between the ports **1** of initiator **1, 104**, devices **1-4, 106, 108, 110, 112,** and initiator **2, 114** are provided through the edge expander **1, 102**.

[0019] In accordance with features of the preferred embodiment, a novel method of interconnecting the components of a SAS domain is provided that significantly improves the fault resiliency of the system, and that is provided without incurring any additional product cost over a standard SAS configuration. The improved fault resiliency is achieved by a novel usage of a subtractive routing port **120** of edge expanders **0, 1, 102** in the SAS network **100** of the preferred embodiment.

[0020] As shown in **FIG. 4**, the subtractive routing ports **120** of edge expanders **0, 1, 102** are connected together in the SAS network **100** of the preferred embodiment. The subtractive routing port **120** of each edge expanders **0, 1, 102** in SAS network **100** allows frames to be sent between nodes not directly attached to the same edge expander. The subtractive routing ports **120** connecting the two edge expanders **0, 1, 102** advantageously are composed of multiple phys for enabling multiple concurrent connections to be established.

[0021] In SAS network **100** of the preferred embodiment, each of the edge expanders **0, 1, 102** is implemented with a conventional device or edge expander typically used to interconnect SAS devices to enable larger topologies, and having behavior as described in the "Serial Attached SCSI Specification" by American National Standards Institute (ANSI). The capabilities provided by the components of the standard SAS configurations of **FIGS. 2 and 3** are the same capabilities utilized to implement the SAS network **100** providing resilient connectivity of the preferred embodiment.

[0022] When a frame is received by the edge expanders **0, 1, 102** at a particular direct routing port, such as direct routing port **0, 118**, the edge expander compares the destination SAS address contained within the frame to the SAS address of each of the nodes of the other direct routing ports **1-5, 118** or SAS address of SAS devices **106, 108, 110, 112,** and **114**. If a match is found, then the frame is routed to that node. If no match is found, then the frame is instead sent to the subtractive routing port **120**. When the subtractive routing port **120** receives a frame, substantially the same edge expander behavior occurs. The expander **102** compares the destination SAS address contained within the frame to the SAS address of each of the nodes of the direct routing ports **0-5, 118** or SAS address of SAS devices **104, 106, 108, 110, 112,** and **114**. An exception is that if a match is not found, then the frame is rejected instead of being resent on the subtractive routing port **120**.

[0023] SAS network **100** of the preferred embodiment improves upon the reliability provided by the standard redundant SAS configuration shown in **FIG. 3**. **FIG. 3** is tolerant of the failure of any single component or link in the SAS domain, however, the concurrent failure of multiple components can cause loss of access between nodes in the domain. SAS network **100** prevents this loss of access by interconnecting the components **104, 106, 108, 110, 112,** and **114** in a novel fashion with the subtractive routing ports **120** of edge expanders **0, 1, 102** and thereby introducing additional paths that may be used.

[0024] In the conventional redundant SAS configuration of **FIG. 3**, there are two paths between each pair of nodes and each path utilizes one of the two ports of the beginning node and connects to a specific port of the end node.

[0025] In SAS network **100** there are four paths between each pair of nodes because each port of the beginning node may be connected to either port of the end node. The additional connections are made possible because the edge expanders **0, 1, 102** are connected together through their respective subtractive routing ports **120**, and enable the formation of the single larger SAS domain of SAS network **100** instead of two smaller SAS domains as in the conventional SAS configuration of **FIG. 3**.

[0026] The type of configuration of SAS network **100** was not possible in earlier technologies, such as the FC-AL configuration illustrated in **FIG. 1**, because the earlier technologies were loop-based. Connecting the two loops together in a configuration such as that depicted in **FIG. 1** would introduce a single point of failure, and in fact the possibility of a single point of failure was what led to providing SAS devices that contain two ports.

[0027] The prior art SAS configuration shown in **FIG. 2** is also composed of a single SAS domain, however this configuration contains single points of failure, for example, in the edge expanders and the fanout expander and does not provide the desired level of reliability and fault tolerance. SAS network **100** achieves the high level of reliability and fault tolerance while eliminating the need for a fanout expander of the prior art SAS configuration of **FIG. 2**.

[**0028**] All components necessary to implement the SAS network **100** of the preferred embodiment are low-cost industry standard components, with no custom hardware components required. As can be seen by comparing the prior art SAS configuration of **FIG. 3** and the SAS network **100** of the preferred embodiment, the same number of components of the same type is required. The only change required to implement the SAS network **100** of the preferred embodiment is to vary how these components are interconnected, and to enable the software support to utilize the additional alternate paths from an initiator to a target. This enables the construction of SAS network **100** of the preferred embodiment for the same cost as a system not utilizing our invention.

[**0029**] The presence of the additional paths between nodes, created by using the subtractive routing ports **120**, also yields better performance characteristics for SAS network **100**. Overall system performance is maintained in the presence of failure, or any single-ported devices **104**, **106**, **108**, **110**, **112**, **114**, while prior art arrangements experienced performance degradation as components failed. This can be understood, for example, by comparing the prior art SAS configuration of **FIG. 3** and SAS network **100**.

[**0030**] First assume that a number of devices are either single-port devices, such as device **3**, **110**, and device **4**, **112**, having only a functional port **0** connected to the edge expander **0**, **102**, as would be the case if Serial ATA devices are attached to the SAS domain using Serial ATA Tunneling Protocol (STP), or are dual-ported devices with a failed port. In the prior art SAS configuration of **FIG. 3** each device is accessible only via a single initiator port through the edge expander to the functional port of the target device. Depending on the characteristics of the workload, this port may become over utilized while the other initiator port is under utilized.

[**0031**] SAS network **100** allows the otherwise under utilized port of the initiator to reach the target device. SAS network **100** does this by reaching the functional port of the target device via the under utilized port, to the edge expander **0** or **1**, **102** not connected to the functioning port of the device, through the subtractive routing port **120** to the other edge expander, and then reaching the functioning port of the device. Had this path not been available, then the initiator would have incurred additional latency and reduced system performance while waiting to use the port connected to the edge expander that was directly connected to the media device.

[**0032**] As illustrated in **FIG. 4**, the subtractive routing port **120** connecting the two edge expanders **0**, **1**,**102** can be composed of multiple phys to enable multiple concurrent connections to be established of this type. This multiple phy subtractive routing port **120** is allowed by the SAS architecture and also desirable in a standard configuration such as shown in **FIG. 2** to enable multiple concurrent connections through the fanout expander. This enables SAS network **100** again to obtain maximum benefit while again utilizing standard components.

[**0033**] In a clustered highly-available configuration, multiple initiator adapters are talking to the storage devices. For the case of a high-function initiator adapter, it may store configuration and state information on reserved areas of the media that are not exposed to the system. This might include, for instance, information on the current state of the RAID array. In a system that is functioning normally one adapter would have primary responsibility for managing the state of these devices and controlling the contents of the configuration and state metadata stored on the device.

[**0034**] In a standard configuration, such as shown in the prior art SAS configuration of **FIG. 3**, there are failure modes where the adapters cannot detect the presence of the other adapter, yet both adapters still have access to all of the storage devices. An example of such a failure mode is to have failures occur in both initiator **1** port **0** and initiator **2** Port **1**. In this case, the adapters may make erroneous decisions and possibly irrevocably corrupt the state of the storage devices. For example both adapters might be concurrently updating the metadata in the devices with conflicting information. Previous solutions, such as those implemented using FC-AL technology as shown in **FIG. 1**, have incorporated additional hardware into the system such that the adapters have additional methods with which to determine the configuration.

[**0035**] Using SAS network **100** this failure mode has been essentially eliminated. That is, the failure mode where the two adapters can each access the same device, but cannot detect each other is eliminated in SAS network **100** through the usage of the additional paths through the subtractive routing ports **120** of the edge expanders **0**, **1**, **102**. Increasing the width of the subtractive routing ports **120** of the edge expanders **0**, **1**, **102** by incorporating additional phys such that a single driver or receiver failure does not cause communication loss, further reduces the likelihood of failure. As described above, these additional phys are also desirable for enabling multiple concurrent connections to be established and are provided in the preferred embodiment of the edge expanders **0**, **1**, **102** of SAS network **100**. SAS network **100** has lower system cost and reduced complexity as compared to prior art arrangements, and additionally does not require the additional development or inclusion of custom hardware to provide the enhanced level of function.

[**0036**] SAS network **100** also allows enhanced reliability for legacy devices that are attached using another storage protocol, the Serial ATA protocol. Serial ATA devices may be attached directly to a SAS domain because SAS is backwards compatible with Serial ATA as is described in the "Serial Attached SCSI Specification" by ANSI, and the Serial ATA devices are then communicated with via the Serial ATA Tunneled Protocol (STP). However, all Serial ATA devices are single-ported and might be attached as shown in **FIG. 2**, or to one of the edge expanders shown in **FIG. 3**. If the initiator port connected to that edge expander fails, then all access is lost to the device. However, if a Serial ATA device is attached to SAS network **100**, then there are two paths to the Serial ATA device since either initiator port may be used to access the Serial ATA device. Either initiator port may fail without losing access to the device. This enables reliability and fault tolerance to be enhanced for a storage device utilizing a technology, such as Serial ATA, that is architected as a direct link without any fault tolerant provisions.

[0037] In brief summary, in accordance with advantages of the preferred embodiment, system fault tolerance is increased. Access is possible from any node to any other node in the presence of multiple failures in the SAS network **100** of the preferred embodiment. This is a significant improvement over prior FC-AL designs, and can be used as a building block in the construction of fault-tolerant autonomic systems. System cost for the fault-tolerant SAS network **100** of the preferred embodiment is substantially the same as that for a standard configuration. SAS configuration **100** utilizes components that are industry standard, low-cost, and readily available. System bandwidth and latency performance is preserved even in the presence of a port failure on a node because the availability of additional paths allow congestion or link saturation to be bypassed, that is, congestion or link saturation is prevented. SAS network **100** of the preferred embodiment also creates multiple paths to a single-ported device, which also enables higher performance by allowing more efficient use of the available bandwidth via bypassing an over utilized link. SAS configuration **100** is readily applicable to the creation of cluster highly available (HA) solutions. Prior art arrangements has possible failure modes where two initiator adapters could not see each other but, both still had maintained access to the media devices. In this case the adapters could make incorrect, and potentially catastrophic, decisions on what were the correct actions. To limit this problem, some prior art systems introduced additional hardware components, resulting in increase system cost and complexity, to detect these failure modes. SAS configuration **100** has eliminated those failure modes so that the additional hardware is not required, and correct determinations are made on the system configuration.

[0038] While the present invention has been described with reference to the details of the embodiments of the invention shown in the drawing, these details are not intended to limit the scope of the invention as claimed in the appended claims.

What is claimed is:

**1**. A method for implementing resilient connectivity in a Serial Attached SCSI (SAS) domain comprising the steps of:

connecting a first edge expander to a first port of a plurality of SAS devices for enabling communications between each of said plurality of SAS devices through said first edge expander;

connecting a second edge expander to a second port of said plurality of SAS devices for enabling communications between each of said plurality of SAS devices through said second edge expander; and

connecting together a subtractive routing port of each of said first edge expander and said second edge expander for enabling communications between each of said plurality of SAS devices via said first ports and said second ports of said plurality of SAS devices.

**2**. A method for implementing resilient connectivity in a Serial Attached SCSI (SAS) domain as recited in claim 1 wherein the step of connecting said first edge expander includes the step of connecting respective direct routing ports of said first edge expander to said first port of said plurality of SAS devices; and wherein the step of connecting said second edge expander includes the step of connecting respective direct routing ports of said second edge expander to said second ports of said plurality of SAS devices.

**3**. A method for implementing resilient connectivity in a Serial Attached SCSI (SAS) domain as recited in claim 1 includes the step of ganging together multiple point-to-point links to form said subtractive routing ports of said first edge expander and said second edge expander for enabling multiple concurrent connections with said subtractive routing ports.

**4**. A method for implementing resilient connectivity in a Serial Attached SCSI (SAS) domain as recited in claim 1 includes the step of utilizing said subtractive routing ports of said first edge expander and said second edge expander for communicating from said first ports of said plurality of SAS devices to said second ports of said plurality of SAS devices.

**5**. A method for implementing resilient connectivity in a Serial Attached SCSI (SAS) domain as recited in claim 1 includes the step of utilizing said subtractive routing ports of said first edge expander and said second edge expander for communicating from said second ports of said plurality of SAS devices to said first ports of said plurality of SAS devices.

**6**. A method for implementing resilient connectivity in a Serial Attached SCSI (SAS) domain as recited in claim 1 includes the step of selectively utilizing said subtractive routing ports of said first edge expander and said second edge expander for workload balancing of communications between said plurality of SAS devices with a failure of one or more of said first and second ports of said plurality of SAS devices.

**7**. Apparatus for implementing resilient connectivity in a Serial Attached SCSI (SAS) domain comprising:

a plurality of SAS devices, each having a first port and a second port;

a first edge expander and a second edge expander, each edge expander having a plurality of direct routing ports and a subtractive routing port;

each of said plurality of direct routing ports of said first edge expander respectively connected to said first port of a respective one of said plurality of SAS devices for enabling communications between each of said plurality of SAS devices;

each of said plurality of direct routing ports of the second edge expander respectively connected to said second port of a respective one of said plurality of SAS devices for enabling communications between each of said plurality of SAS devices; and

said subtractive routing ports of said first edge expander and said second edge expander connected together for enabling communications between each of said plurality of SAS devices via said first ports and said second ports of said plurality of SAS devices.

**8**. Apparatus for implementing resilient connectivity in a Serial Attached SCSI (SAS) domain as recited in claim 7 wherein said subtractive routing ports of said first edge expander and said second edge expander include multiple point-to-point links enabling multiple concurrent connections.

**9**. Apparatus for implementing resilient connectivity in a Serial Attached SCSI (SAS) domain as recited in claim 7 wherein said subtractive routing ports of said first edge expander and said second edge expander are used for communicating from said first ports of said plurality of SAS devices to said second ports of said plurality of SAS devices.

**10**. Apparatus for implementing resilient connectivity in a Serial Attached SCSI (SAS) domain as recited in claim 7 wherein said subtractive routing ports of said first edge expander and said second edge expander are used for communicating from said second ports of said plurality of SAS devices to said first ports of said plurality of SAS devices.

**11**. A Serial Attached SCSI (SAS) network for implementing resilient connectivity in a SAS domain comprising:

a first edge expander and a second edge expander, each edge expander having a plurality of direct routing ports and a subtractive routing port;

each of said plurality of direct routing ports of said first edge expander respectively connected to said first port of a respective one of a plurality of SAS devices for enabling communications between each of said plurality of SAS devices;

each of said plurality of direct routing ports of the second edge expander respectively connected to a second port of a respective one of said plurality of SAS devices for enabling communications between each of said plurality of SAS devices; and

said subtractive routing ports of said first edge expander and said second edge expander connected together for enabling communications between each of said plurality of SAS devices via said first ports and said second ports of said plurality of SAS devices.

**12**. A Serial Attached SCSI (SAS) network as recited in claim 11 wherein said subtractive routing ports of said first edge expander and said second edge expander include multiple point-to-point links enabling multiple concurrent connections.

* * * * *

US 20020181476A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2002/0181476 A1**

**Badamo et al.** (43) **Pub. Date: Dec. 5, 2002**

(54) **NETWORK INFRASTRUCTURE DEVICE FOR DATA TRAFFIC TO AND FROM MOBILE UNITS**

(76) Inventors: **Michael J. Badamo**, Rockville, MD (US); **David G. Barger**, Darnestown, MD (US); **Tony M. Cantrell**, Hagerstown, MD (US); **Wayne McNinch**, Monrovia, MD (US); **Christopher C. Skiscim**, Dickerson, MD (US); **David M. Summers**, Gaithersburg, MD (US); **Peter Szydlo**, Whistling Wind Way, MD (US)

Correspondence Address:
**MCGLEW & TUTTLE, PC**
**SCARBOROUGH STATION**
**SCARBOROUGH, NY 10510 (US)**

(21) Appl. No.: **09/811,204**

(22) Filed: **Mar. 17, 2001**

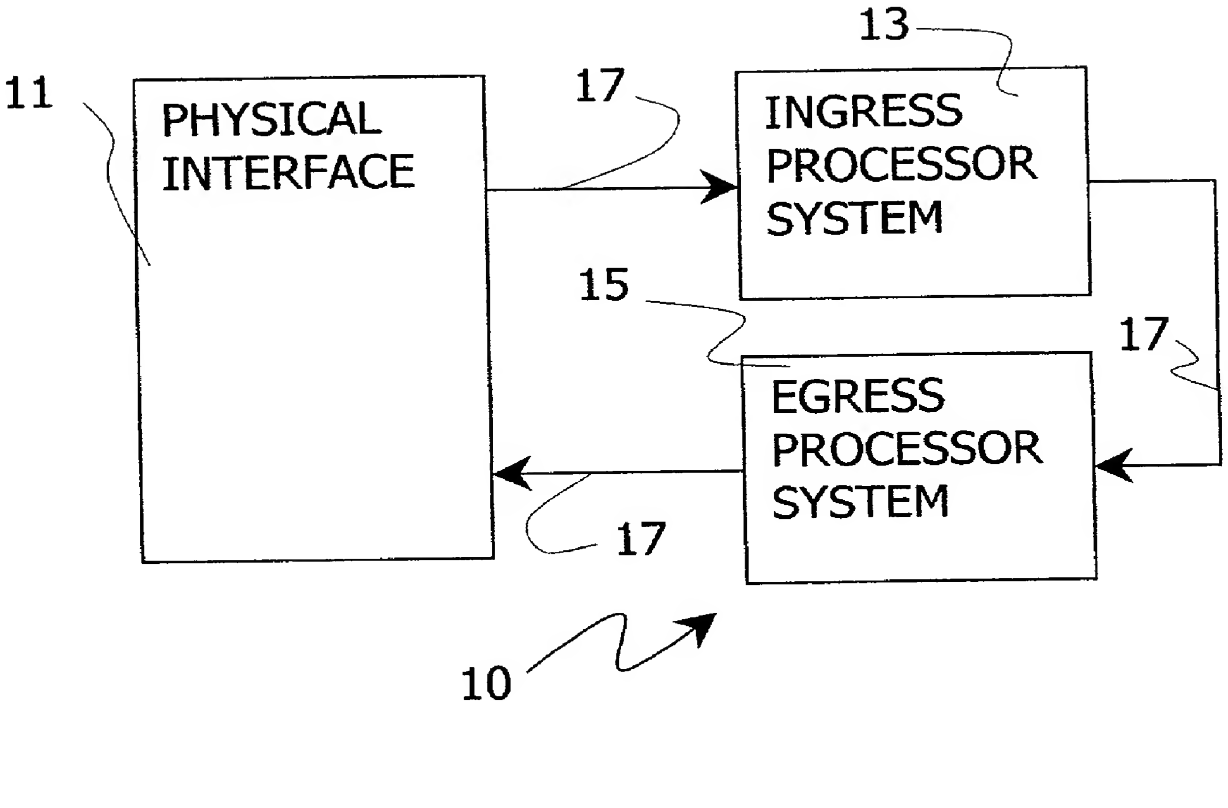
**Publication Classification**

(51) **Int. Cl.**$^{7}$ .................................................... **H04L 12/28**

(52) **U.S. Cl.** .......................... **370/401**; 370/389; 370/352

(57) **ABSTRACT**

A network gateway device has a physical interface for connection to a medium. The device has an ingress processor system for ingress processing of all or part of packets received from the physical interface and for sending ingress processed packets for egress processing. The device has an egress processor system for receiving ingress processed packets and for egress processing of all or part of received packets for sending to physical interface. Interconnections are provided, including an interconnection between the ingress processor and the egress processor, including an interconnection between the ingress processor and the physical interface, and including an interconnection between the ingress processor and the physical interface. A packet queue is provided with packets awaiting transmission. The packet queue may be the exclusive buffer for packets between packets entering the device and packet transmission. The packets may exit the device at a rate of the line established at the physical interface. The ingress processing system processes packets including at least one or more of protocol translation, de-encapsulation, decryption, authentication, point-to-point protocol (PPP) termination and network address translation (NAT). The egress processing system processes packets including at least one or more of protocol translation, encapsulation, encryption, generation of authentication data, PPP generation and NAT.



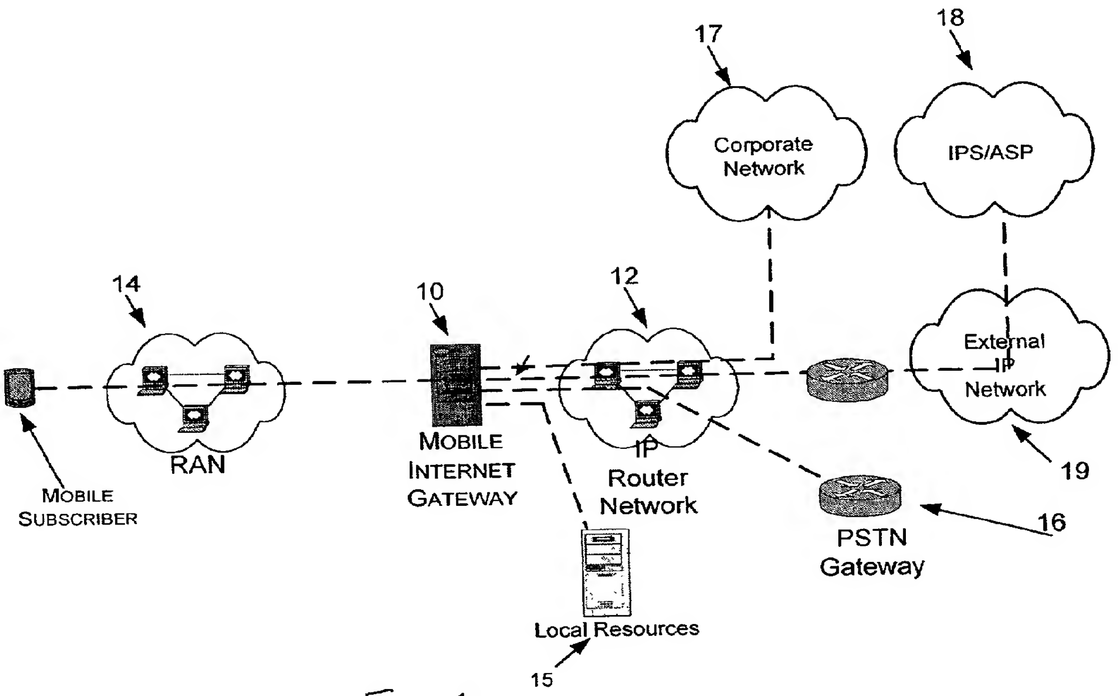

17
18
Corporate Network
IPS/ASP
14
10
12
External IP Network
RAN
MOBILE INTERNET GATEWAY
IP Router Network
19
MOBILE SUBSCRIBER
16
PSTN Gateway
Local Resources
15

Fig. 1a

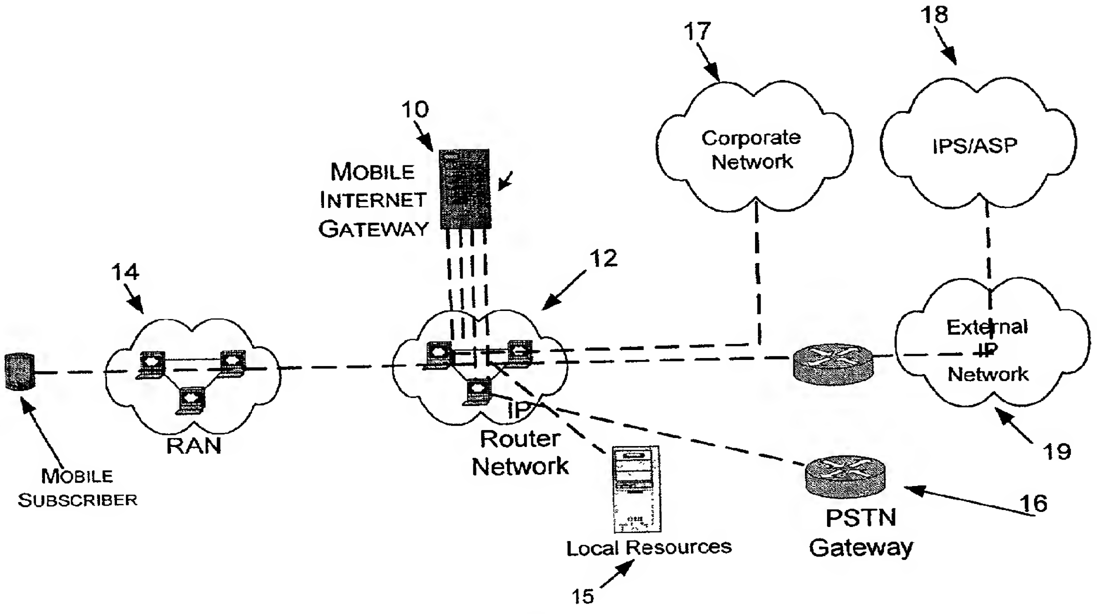

17
18
10
Corporate Network
IPS/ASP
MOBILE INTERNET GATEWAY
12
14
External IP Network
RAN
IP Router Network
19
MOBILE SUBSCRIBER
16
PSTN Gateway
Local Resources
15

Fig. 1b

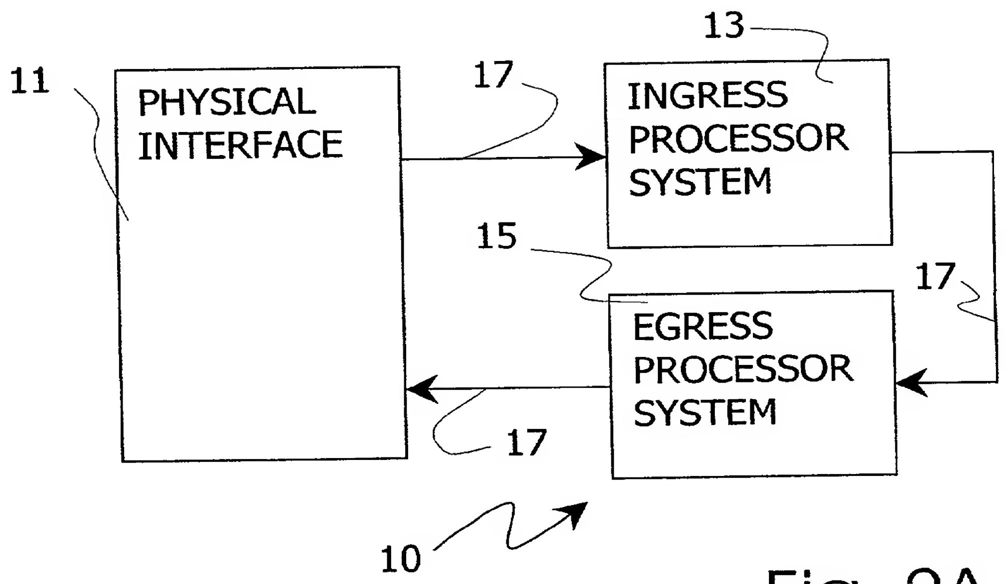

13
11
PHYSICAL INTERFACE
17
INGRESS PROCESSOR SYSTEM
15
17
EGRESS PROCESSOR SYSTEM
17
10

Fig. 2A

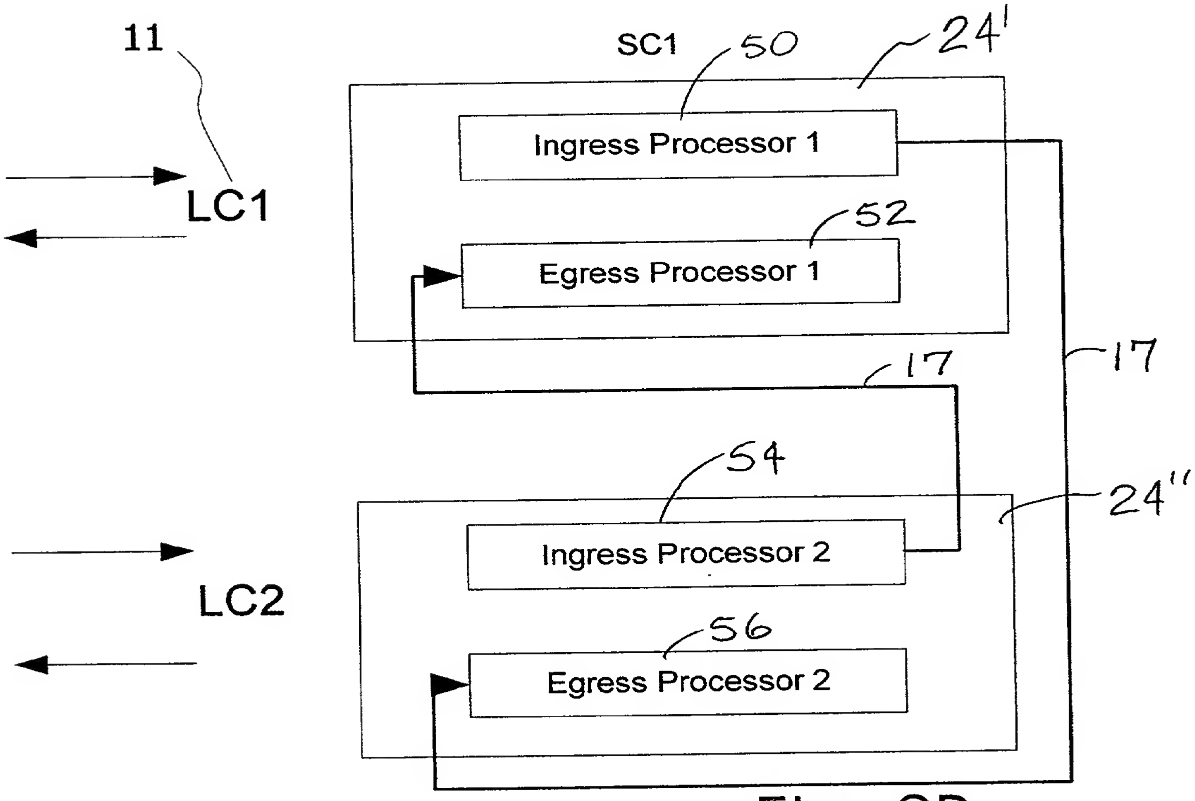

11
SC1
50
24'
Ingress Processor 1
LC1
52
Egress Processor 1
17
17
54
24''
Ingress Processor 2
LC2
56
Egress Processor 2

Fig. 2B

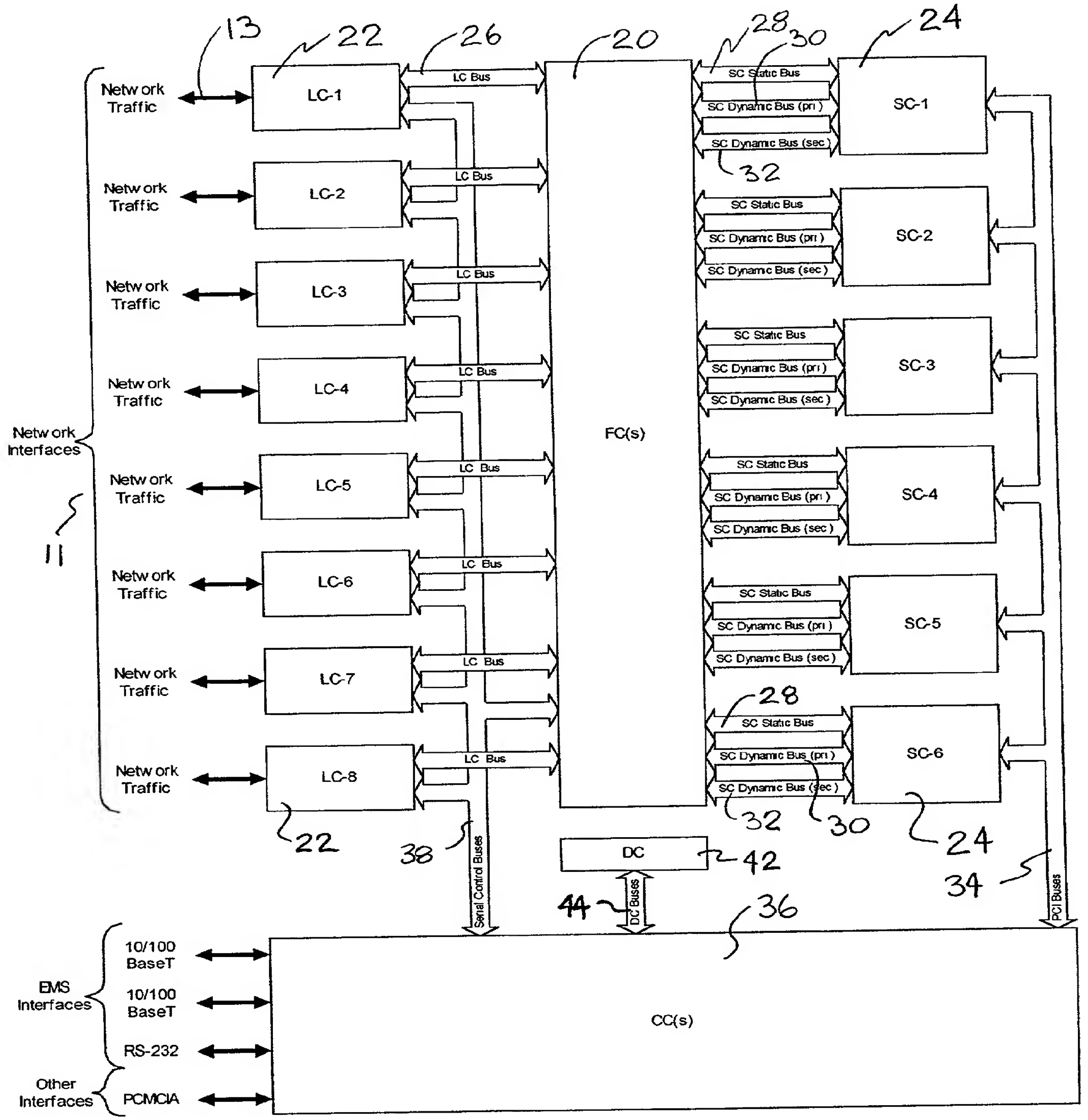

13
22
26
20
28
30
24
Network Traffic
LC-1
LC-2
LC-3
LC-4
LC-5
LC-6
LC-7
LC-8
LC Bus
Network Interfaces
11
FC(s)
SC Static Bus
SC Dynamic Bus (pri)
SC Dynamic Bus (sec)
SC-1
SC-2
SC-3
SC-4
SC-5
SC-6
32
38
Serial Control Buses
DC
42
44
DC Buses
36
34
PCI Buses
EMS Interfaces
10/100 BaseT
10/100 BaseT
RS-232
Other Interfaces
PCMCIA
CC(s)

Fig. 3

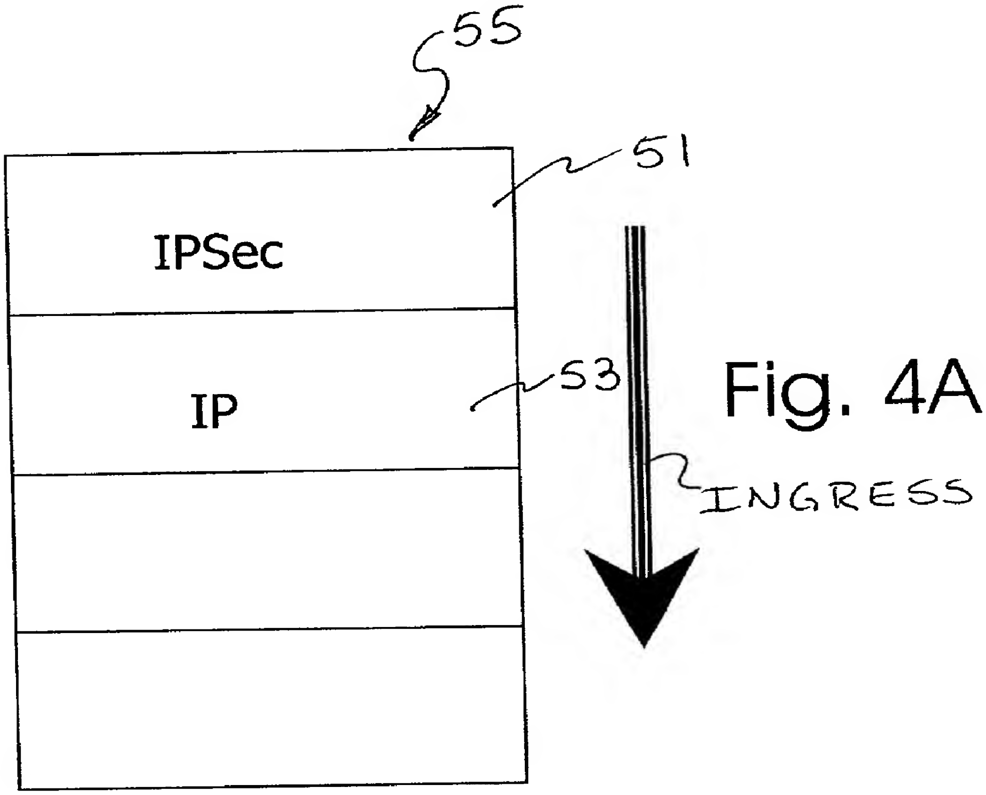


Fig. 4A

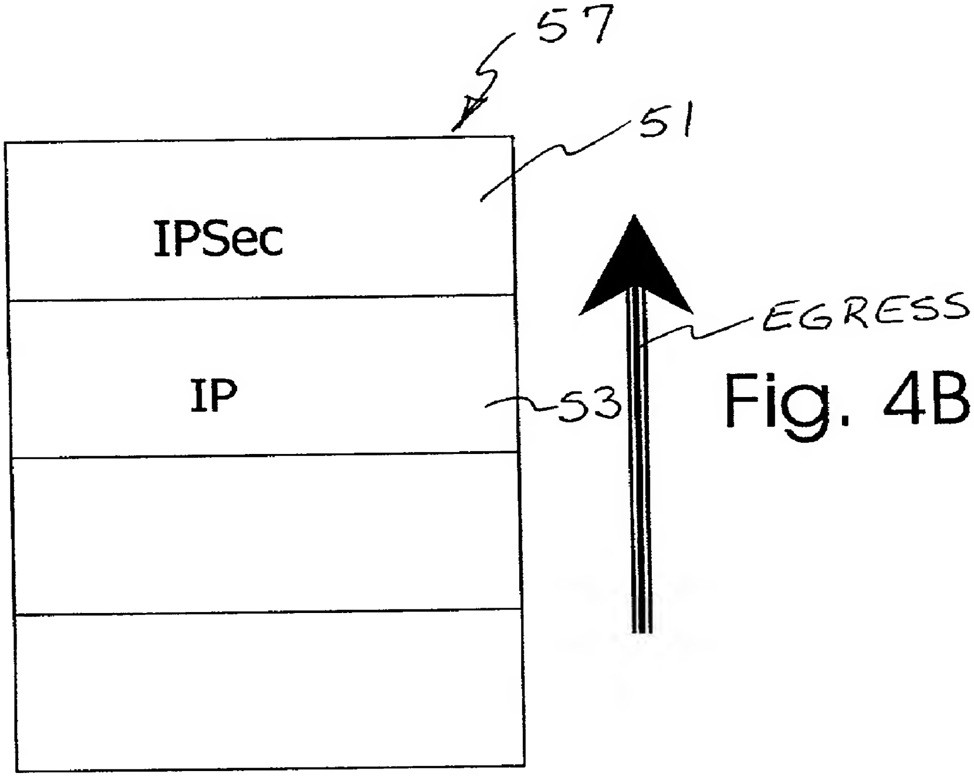


Fig. 4B

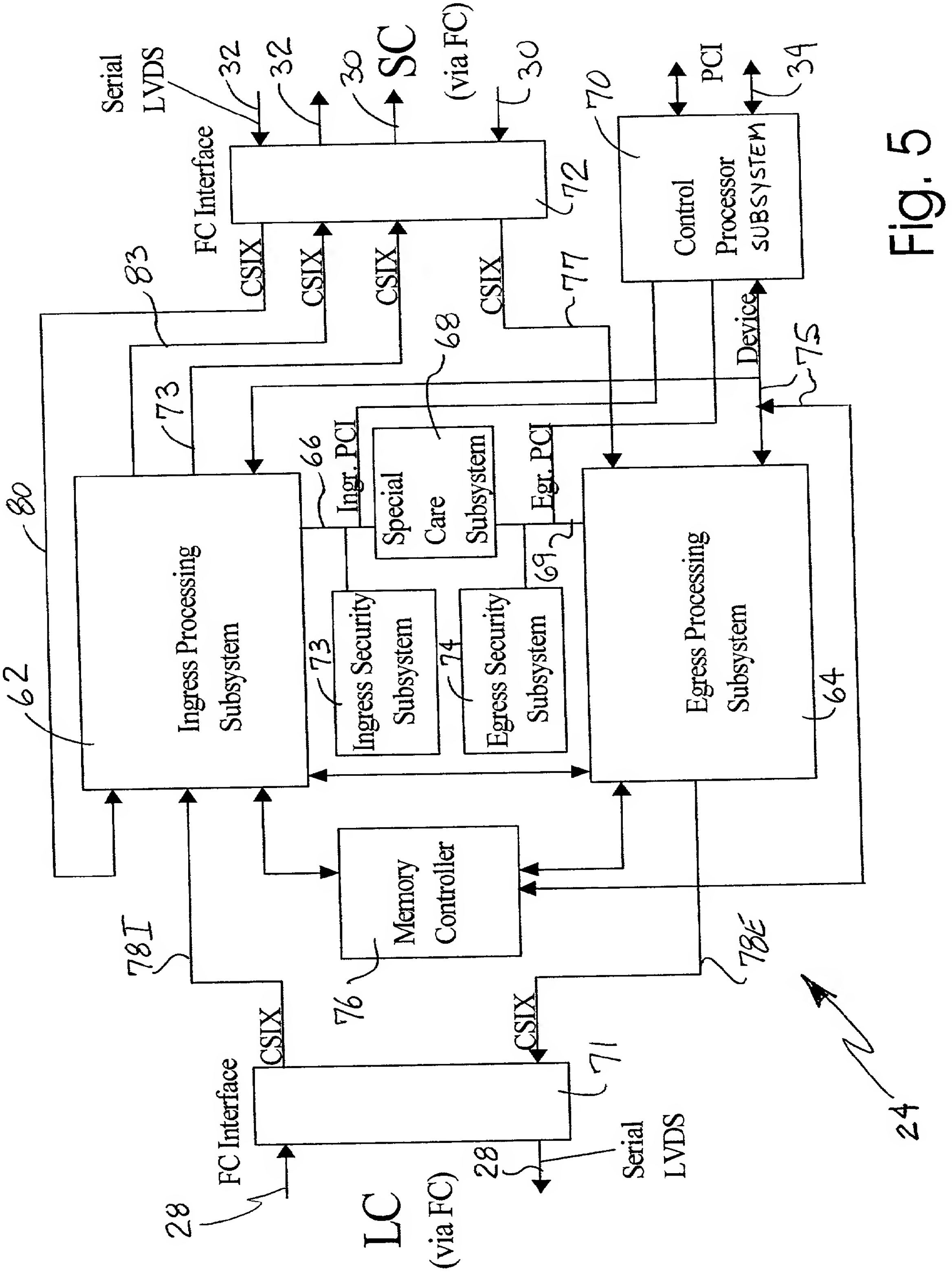

Serial
LVDS
32
32
30
SC
(via FC)
30
FC Interface
72
70
PCI
34
Control
Processor
SUBSYSTEM
CSIX
CSIX
CSIX
CSIX
83
77
68
Device
75
73
66
Ingr. PCI
Special
Care
Subsystem
Egr. PCI
69
80
Ingress Processing
Subsystem
62
73
Ingress Security
Subsystem
74
Egress Security
Subsystem
Egress Processing
Subsystem
64
Memory
Controller
76
78I
78E
CSIX
CSIX
71
FC Interface
28
Serial
LVDS
28
LC
(via FC)
24

Fig. 5

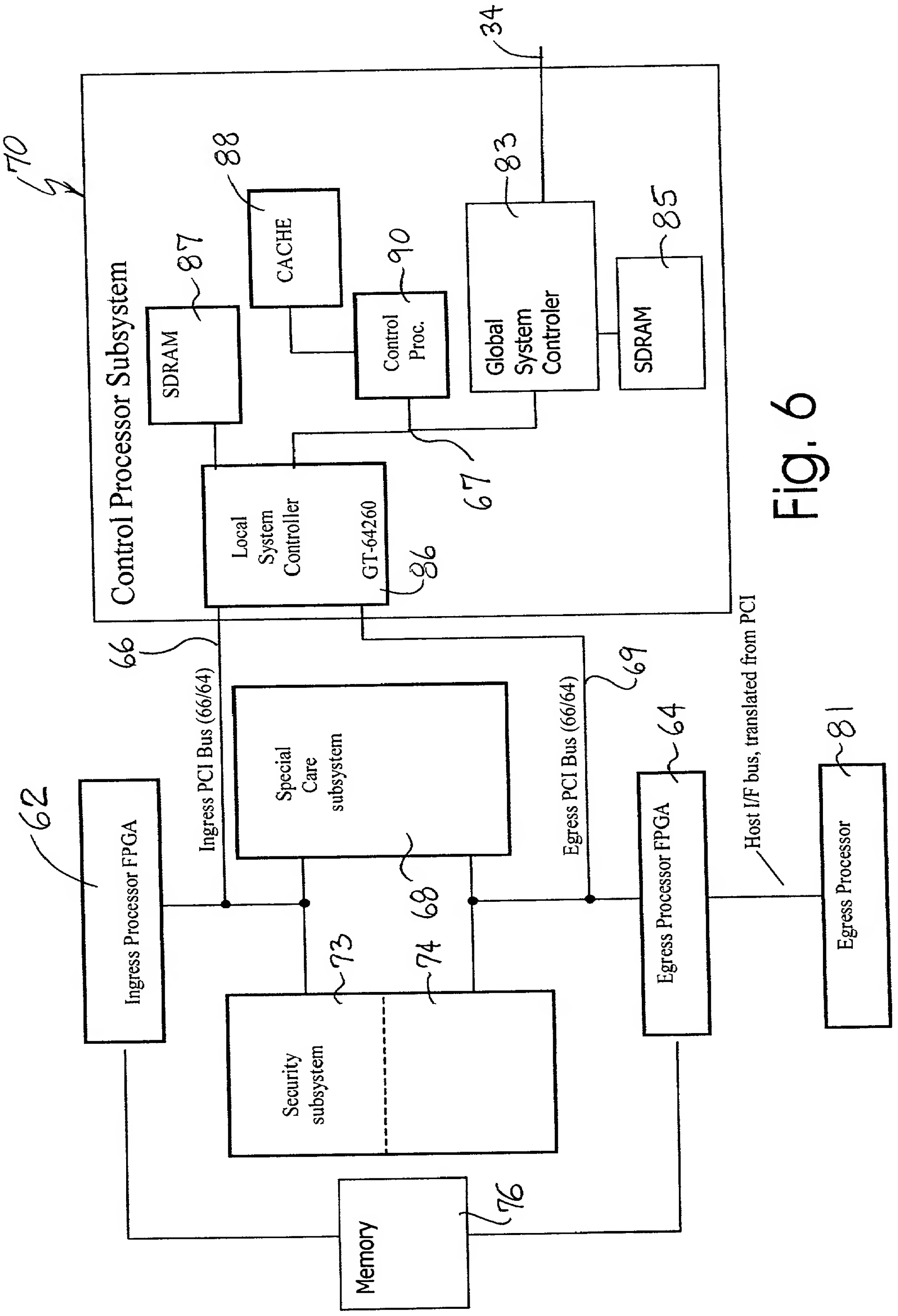

Control Processor Subsystem
70
34
88
CACHE
87
SDRAM
90
Control Proc.
83
Global System Controler
85
SDRAM
Local System Controller
GT-64260
86
67
66
Ingress PCI Bus (66/64)
Special Care subsystem
68
69
Egress PCI Bus (66/64)
64
Host I/F bus, translated from PCI
81
Egress Processor
Egress Processor FPGA
62
Ingress Processor FPGA
73
74
Security subsystem
Memory
76

Fig. 6

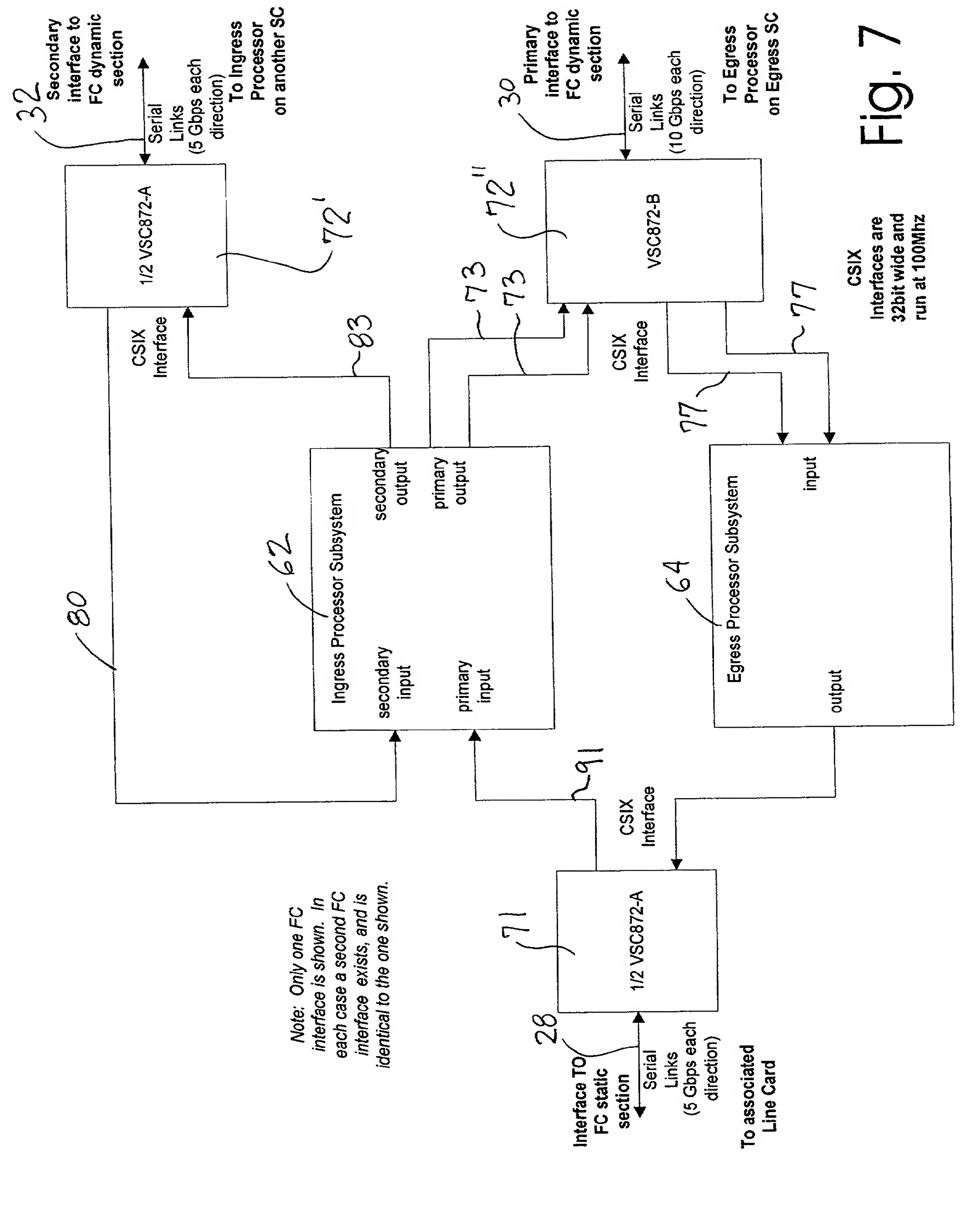

32
Secondary interface to FC dynamic section
Serial Links (5 Gbps each direction)
To Ingress Processor on another SC
1/2 VSC872-A
72'
CSIX Interface
83
80
30
Primary interface to FC dynamic section
Serial Links (10 Gbps each direction)
To Egress Processor on Egress SC
VSC872-B
72''
73
73
CSIX Interface
77
77
CSIX Interfaces are 32bit wide and run at 100Mhz
Ingress Processor Subsystem
62
secondary output
primary output
secondary input
primary input
Egress Processor Subsystem
64
input
output
91
CSIX Interface
1/2 VSC872-A
71
28
Interface TO FC static section
Serial Links (5 Gbps each direction)
To associated Line Card
Note: Only one FC interface is shown. In each case a second FC interface exists, and is identical to the one shown.

Fig. 7

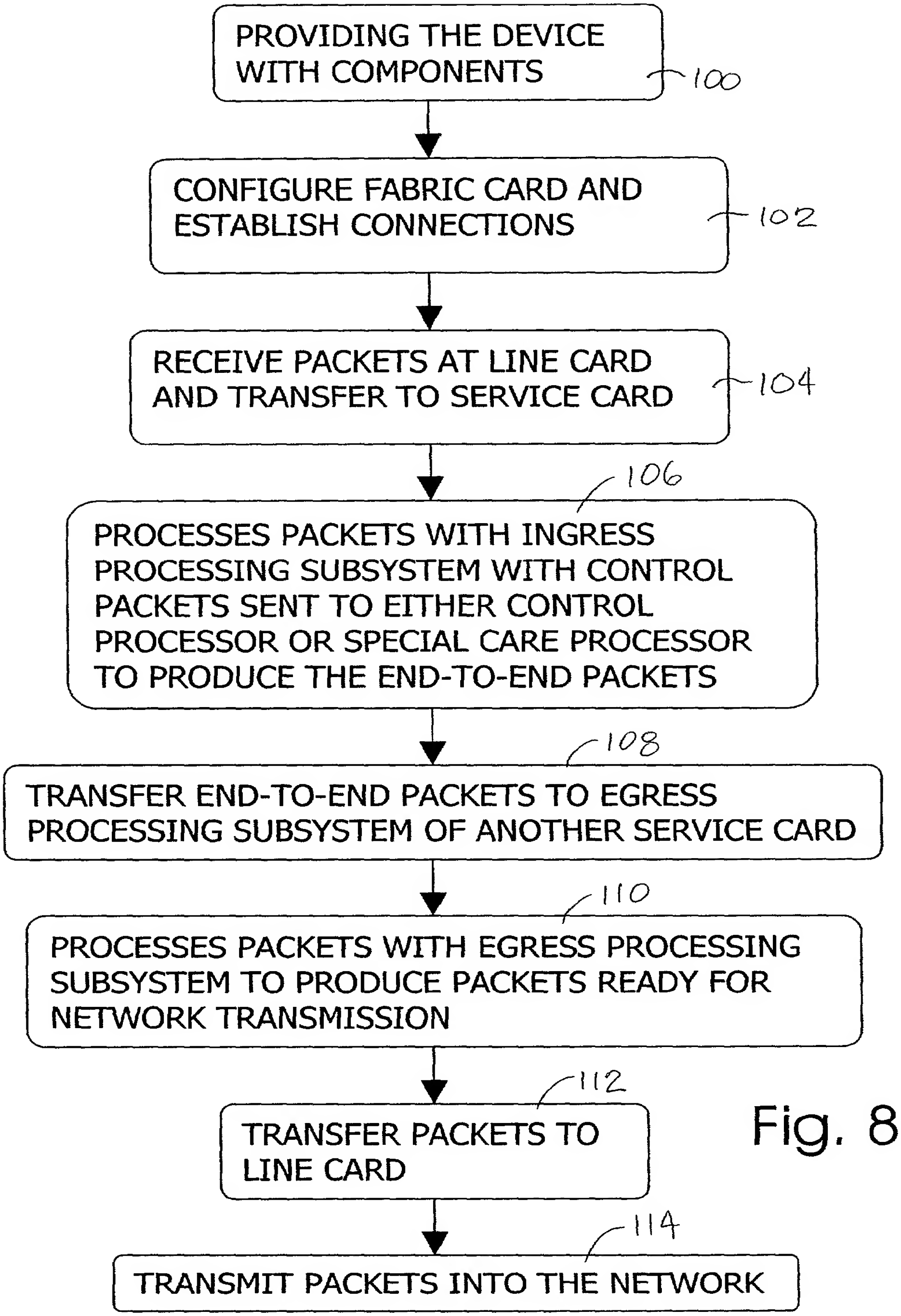

PROVIDING THE DEVICE WITH COMPONENTS
100
CONFIGURE FABRIC CARD AND ESTABLISH CONNECTIONS
102
RECEIVE PACKETS AT LINE CARD AND TRANSFER TO SERVICE CARD
104
106
PROCESSES PACKETS WITH INGRESS PROCESSING SUBSYSTEM WITH CONTROL PACKETS SENT TO EITHER CONTROL PROCESSOR OR SPECIAL CARE PROCESSOR TO PRODUCE THE END-TO-END PACKETS
108
TRANSFER END-TO-END PACKETS TO EGRESS PROCESSING SUBSYSTEM OF ANOTHER SERVICE CARD
110
PROCESSES PACKETS WITH EGRESS PROCESSING SUBSYSTEM TO PRODUCE PACKETS READY FOR NETWORK TRANSMISSION
112
TRANSFER PACKETS TO LINE CARD
114
TRANSMIT PACKETS INTO THE NETWORK

Fig. 8

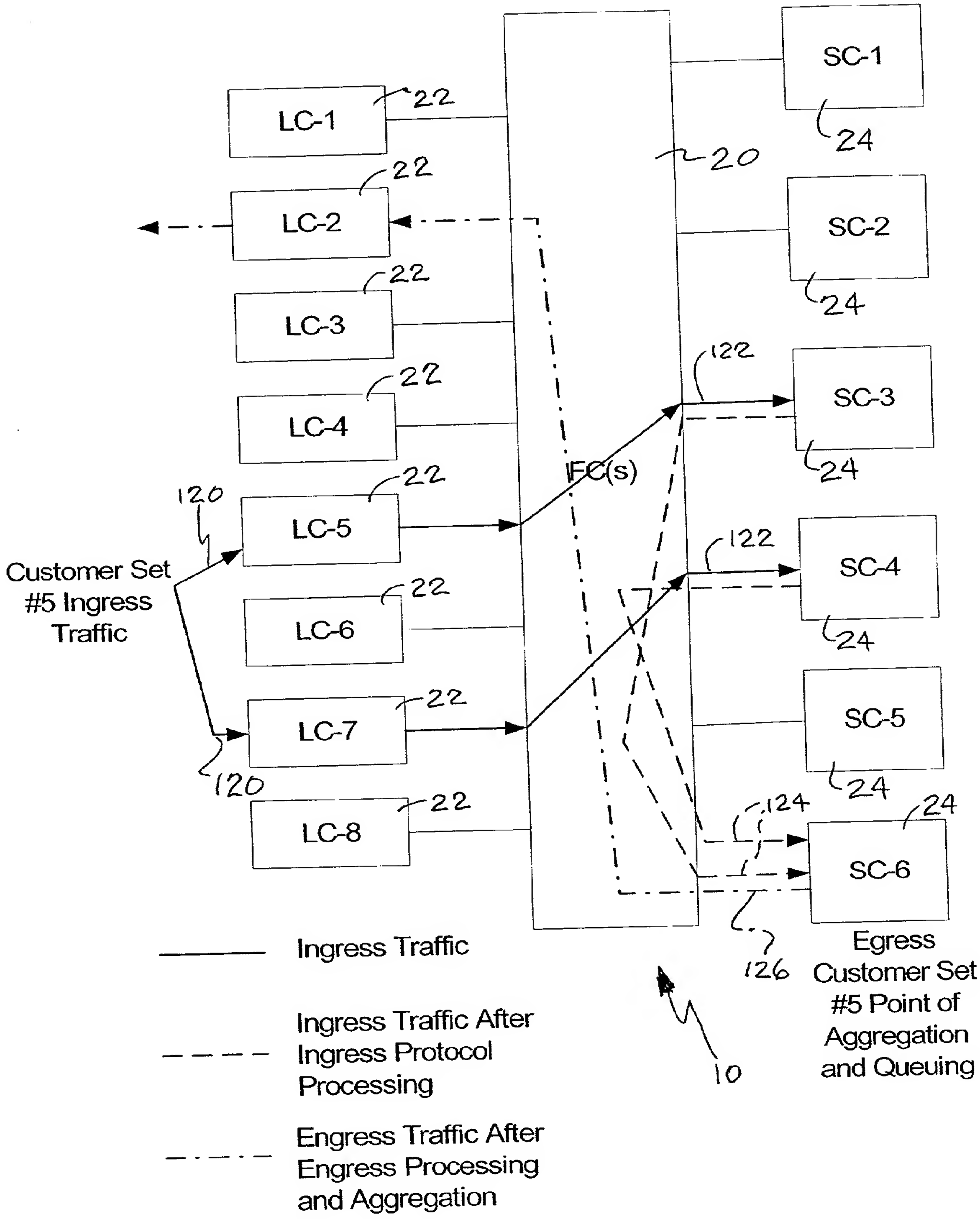

LC-1
LC-2
LC-3
LC-4
LC-5
LC-6
LC-7
LC-8
22
IFC(s)
20
SC-1
SC-2
SC-3
SC-4
SC-5
SC-6
24
122
124
126
120
10
Customer Set #5 Ingress Traffic
Egress Customer Set #5 Point of Aggregation and Queuing
Ingress Traffic
Ingress Traffic After Ingress Protocol Processing
Engress Traffic After Engress Processing and Aggregation

FIG. 9

# NETWORK INFRASTRUCTURE DEVICE FOR DATA TRAFFIC TO AND FROM MOBILE UNITS

## FIELD OF THE INVENTION

**[0001]** The present invention generally relates to the mobile Internet and more particularly relates to network infrastructure devices such as mobile Internet gateways that allow wireless data communication users to access content through the Internet protocol (IP). The invention also relates to a process by which users of the IP network (or users connected through the IP network) can communicate with users of wireless data communications devices.

## BACKGROUND OF THE INVENTION

**[0002]** In order for users of wireless data communications devices to access content on or through the IP, a gateway device is required that provides various access services and subscriber management. Such a gateway also provides a means by which users on the IP network (or connected through the IP network) can communicate with users of wireless data communications devices.

**[0003]** The architecture of such a device must adhere to and process the mobile protocols, be scalable and reliable, and be capable of flexibly providing protocol services to and from the IP network. Traffic arriving from, or destined for the IP Router Network (e.g. the Internet) can use a variety of IP-based protocols, sometimes in combination. The device should also be able to provide protocol services to the radio access network (RAN) and to the IP Network, scale to large numbers of users without significant degradation in performance and provide a highly reliable system.

**[0004]** Devices have been used that include line cards directly connected to a forwarding device connected to the bus and a control device connected to the bus. The forwarding device performers transmit, receive, buffering, encapsulation, de-encapsulation and filtering. In such arrangement the forwarding device performs all processes related to layer two tunnel traffic. All forwarding decisions, as to an ingress processing (including de-encapsulation, decryption, etc.), are made in one location. Given the dynamics of a system requiring access by multiple users and the possible transfer of large amounts of data, such a system must either limit the number of users, to avoid data processing bottlenecks, or the system must seek faster and faster processing with faster and higher volume buses.

## SUMMARY AND OBJECTS OF THE INVENTION

**[0005]** It is an object of the invention to provide a network device, particularly a gateway device with an ingress processor system for ingress processing of all or part of received packets which is at least partially separate from an egress processor system for receiving ingress processed packets and for egress processing of all or part of received packets whereby packet processing is efficiently handled.

**[0006]** It is another object of the invention to provide a network infrastructure device, particularly for handling traffic arriving from or destined to RAN users, including users of a data communications protocol [s] specific to mobile and RAN technology and for handling traffic arriving from, or destined to the IP router network (e.g. the Internet) in which the system architecture of the device provides protocol services to the RAN and the IP network and is able to scale to large numbers of users without processing or transfer bottlenecks, without significant degradation in performance while providing a highly reliable device.

**[0007]** Is a further object of the invention to provide a network gateway device for communications back and forth between RAN technology and IP network systems providing protocol services for handling traffic between the systems and for processing packets from line cards connected as part of the gateway device with ingress packet processing at least partially physically separate from egress packet processing.

**[0008]** According to the invention, a network gateway device is provided with a physical interface for connection to a medium. The device includes an ingress processor system for ingress processing of all or part of packets received from the physical interface and for sending ingress processed packets for egress processing. The device also includes an egress processor system for receiving ingress processed packets and for egress processing of all or part of received packets for sending to the physical interface. Interconnections are provided including an interconnection between the ingress processor system and the egress processor system, an interconnection between the ingress processor system and the physical interface and an interconnection between the egress processor system and the physical interface.

**[0009]** Advantageously, the device may have a single packet queue establishing a queue of packets awaiting transmission. The packet queue may be the exclusive buffer for packets between packets entering the device and packet transmission. The device allows packets to exit the device at a rate of the line established at the physical interface.

**[0010]** The ingress processing system processes packets including at least one or more of protocol translation, de-encapsulation, decryption, authentication, point-to-point protocol (PPP) termination and network address translation (NAT). The egress processing system processes packets including at least one or more of protocol translation, encapsulation, encryption, generation of authentication data, PPP generation and NAT.

**[0011]** The ingress and egress processor systems may advantageously respectively include a fast path processor subsystem processing packets at speeds greater than or equal to the rate at which they enter the device. The fast path processor system may provide protocol translation processing converting packets from one protocol to another protocol. Each of the ingress and egress processor system may also include a security processor subsystem for processing security packets requiring one or more of decryption and authentication, the processing occurring concurrently with fast path processor packet processing. The processor systems may also include a special care packet processor for additional packet processing concurrently with fast path processor packet processing. The special care packet processor preferably processes packets including one or more of network address translation (NAT) processing and NAT processing coupled with application layer processing (NAT-ALG). The processor systems may also include a control packet processor for additional packet processing concurrently with fast path processor packet processing, including processing packets signaling the start and end of data sessions, packets used to convey information to a particular protocol and packets dependent on interaction with external entities.

[**0012**] The physical interface may include one or more line cards. The ingress processor system may be provided as part of a service card. The egress processor system may be provided as part of the service card or as part of another service card. Such a card arrangement may be interconnected with a line card bus connected to the line card, a service card bus connected to at least one of the service card and the another service card and a switch fabric connecting the line card to at least one of the service card and the another service card. The switch fabric may be used to connect any one of the line cards to any one of the service cards, whereby any line card can send packet traffic to any service card and routing of packet traffic is configured one of statically and dynamically by the line card. The service card bus may include a static bus part for connection of one of the service cards through the switch fabric to one of the line cards and a dynamic bus for connecting a service card to another service card through the fabric card. This allows any service card to send packet traffic requiring ingress processing to any other service card for ingress processing and allowing any service card to send traffic requiring egress processing to any other service card for egress processing. With this the system can make use of unused capacity that may exist on other service cards.

[**0013**] According to another aspect of the invention, a gateway process is provided including receiving packets from a network via a physical interface connected to a medium. The process includes the ingress processing of packets with an ingress processing system. This processing includes one or more of protocol translation processing, de-encapsulation, decryption, authentication, point-to-point protocol (PPP) termination and network address translation (NAT). The packets are then transferred to an egress packet processing subsystem. The process also includes the egress processing of the packets with an egress processing system. The processing includes one or more of protocol translation, encapsulation, encryption, generation of authentication data, PPP generation and NAT processing.

[**0014**] The line cards can be for various media and protocols. The line cards may have one or multiple ports. One or more of the line cards may be a gigabit Ethernet module, an OC-12 module or modules for other media types such as a 155-Mbps ATM OC-3c Multimode Fiber (MMF) module, a 155-Mbps ATM OC-3c Single-Mode Fiber (SMF) module, a 45-Mbps ATM DS-3 module, a **10/100**-Mbps Ethernet I/O module, a 45-Mbps Clear-Channel DS-3 I/O module, a 52-Mbps HSSI I/O module, a 45-Mbps Channelized DS-3 I/O module, a 1.544-Mbps Packet T1 I/O module and others.

[**0015**] The various features of novelty which characterize the invention are pointed out with particularity in the claims annexed to and forming a part of this disclosure. For a better understanding of the invention, its operating advantages and specific objects attained by its uses, reference is made to the accompanying drawings and descriptive matter in which preferred embodiments of the invention are illustrated.

## BRIEF DESCRIPTION OF THE DRAWINGS

[**0016**] In the drawings:

[**0017**] **FIG. 1A** is a schematic drawing of a system using the device according to the invention;

[**0018**] **FIG. 1A** is a schematic drawing of another system using the device according to the invention;

[**0019**] **FIG. 2A** is a diagram showing a processing method and system according to the invention;

[**0020**] **FIG. 2B** is a diagram showing further processing aspects of the processing method shown in **FIG. 2A**;

[**0021**] **FIG. 3** is a diagram showing system components of an embodiment of the device according to the invention;

[**0022**] **FIG. 4A** is a schematic representation of ingress protocol stack implementation, enabling processing of packets to produce an end to end packet (i.e. tunnels are terminated, IPSec packets are decrypted);

[**0023**] **FIG. 4B** is a schematic representation of egress protocol stack implementation, enabling processing of packets including necessary encapsulation and encryption;

[**0024**] **FIG. 5** is a diagram showing service card architecture according to an embodiment of the invention;

[**0025**] **FIG. 6** is a diagram showing the peripheral component interconnect (PCI) data bus structure of a service card according to the embodiment of **FIG. 5**;

[**0026**] **FIG. 7** is a diagram showing the common switch interface (CSIX) data bus structure of a service card according to the embodiment of **FIG. 5**;

[**0027**] **FIG. 8** is a flow diagram showing a process according to the invention; and

[**0028**] **FIG. 9** is a diagram showing single point of queuing features of the invention.

## DESCRIPTION OF THE PREFERRED EMBODIMENTS

[**0029**] Referring to the drawings in particular, the invention comprises a network infrastructure device or mobile Internet gateway **10** as well as a method of communication using the gateway **10**. **FIGS. 1A and 1B** depict two possible deployments of the invention. The invention can form a separation point between two or more networks, or belong to one or more networks. Gateway **10** handles data traffic to and from mobile subscribers via RAN **14**. As shown in **FIG. 1** data traffic arriving from, or destined to users on the RAN **14** must use one or more data communications protocols specific to mobile users and the RAN technology. Traffic arriving from, or destined for the IP Router Network (e.g. the Internet) **12** can use a variety of IP-based protocols, sometimes in combination. The architecture of the gateway **10** described here, the Packet Gateway Node (PGN) **10** solves the problem of being able to provide protocol services to the RAN **14** and to the IP Network **12**, scale to large numbers of users without significant degradation in performance and provide a highly reliable system. It also provides for management of mobile subscribers (e.g., usage restrictions, policy enforcement) as well as tracking usage for purposes of billing and/or accounting.

[**0030**] The IP router network generally designated **12** may include connections to various different networks. The IP router network **12**, for example, may include the Internet and may have connections to external Internet protocol networks **19** which in turn provide connection to Internet service provider/active server pages **18** or which may also provide a connection to a corporate network **17**. The IP router network **12** may also provide connections to the public switched telephone network (PSTN) gateway **16** or for example local resources (data storage etc.) **15**. The showing of **FIGS. 1A and 1B** is not meant to be all inclusive. Other networks and network connections of various different protocols may be provided. The PGN**10** may provide communications between one or more of the networks or provide communications between users of the same network.

[**0031**] It is often the case that the amount of ingress processing differs from egress processing. For example, a request sent for Web content might be very small (with a small amount of ingress processing and a small amount of egress processing). However, the response might be extremely large (i.e., music file etc.). This may require a great deal of ingress processing and a great deal of egress processing. The serial handling of the ingress and egress processing for both the request and the response for a line card (for a particular physical interface connection) may cause problems such as delays. That is, when ingress and egress processing are performed serially, e.g., in the same processor or serially with multiple processors, traffic awaiting service can suffer unpredictable delays due to the asymmetric nature of the data flow.

[**0032**] **FIG. 2A** shows an aspect of the PGN **10** and of the method of the invention whereby the ingress processing and egress processing are divided among different processing systems. Packets are received at the PGN **10** at physical interface **11** and packets are transmitted from the PGN **10** via the physical interface **11**. The physical interface **11** may be provided as one or more line cards **22** as discussed below. An ingress processing system **13** is connected to the physical interface **11** via interconnections **17**. The ingress processing system **13** preforms the ingress processing of received packets. This ingress processing of packets includes at least one or more of protocol translation, de-encapsulation, decryption, authentication, point-to-point protocol (PPP) termination and network address translation (NAT). An egress processing system **15** is connected to the physical interface **11** via interconnections **17** and is also connected to the ingress processing system **13** by interconnections **17**. The egress processing system **13** preforms the ingress processing of received packets. This egress processing of packets includes at least one or more of protocol translation, encapsulation, encryption, generation of authentication data, PPP generation and NAT. The ingress processor **13** and egress processor **15** may be provided as part of a device integrated with the physical interface. Additionally, the ingress processor **13** and egress processor **15** may be provided as part of one or more service cards **24** connected to one or more line cards **22** via the interconnections **17**. The processing method and arrangement allows ingress and egress processing to proceed concurrently.

[**0033**] As shown in **FIG. 2B** one service card **24**' may provide the ingress processing and another service card **24**" may provide the egress processing. The ingress processing or egress processing may be distributed between more than one service card **24**. As shown in **FIG. 2B** a service card **24**' includes ingress processor system **50** and egress processor system **52**. Packets are received from a line card LC1 designated **22**' and packets enter the ingress processor **50** where they are processed to produce end-to-end packets, i.e., tunnels (wherein the original IP packet header is encapsulated) are terminated, Internet protocol security (IPSec) packets are decrypted, Point-to-Point Protocol (PPP) is terminated and NAT or NAT-ALG is performed. The end-to-end packets are then sent to another service card **24**" via interconnections **17**. At this other service card **24**" the egress processor system **56** encapsulates and encrypts the end-to-end packets and the packets are then sent to the LC2 designated **22**" for transmission into the network at interface **11**.

[**0034**] Each of the processor systems **13** and **15** in the example of **FIG. 2A and 50**, **52**, **54** and **56** in the example of **FIG. 2B** is preferably provided with purpose built processors. This allows the processing of special packets, security packets, control packets and simple protocol translation concurrently. This allows the PGN **10** to use a single point of queuing for the device. A packet queue establishes a queue of packets awaiting transmission. This packet queue is the exclusive buffer for packets between packets entering the device and packet transmission. The packets exit the device or complete processing at a rate of the line established at the physical interface (at the rate of the packet ingress). Each processor system preferably includes a fast path processor subsystem processing packets at speeds greater than or equal to the rate at which they enter the device. The fast path processor system provides protocol translation processing converting packets from one protocol to another protocol. Each processor preferably includes a security processor subsystem for processing security packets and preferably a control subsystem for control packets and a special care subsystem for special care packets. The processor subsystems process concurrently. The device allows context (information related to user traffic) to be virtually segregated from other context. Further, the use of multiple service cards allows context to be physically segregated, if this is required.

[**0035**] **FIG. 3** shows a diagram of an embodiment of the hardware architecture. The system architecture of device **10** divides packet processing from traffic to and from the line cards (LCs) **22** via a switch fabric or fabric card (FC) **20**. Processing is performed in service cards (SC) **24**. The LCs **22** are each connected to the FC **20** via a LC bus **26** (static LC bus). The SCs **24** are connected by a SC static bus **28**, SC dynamic bus (primary) **30** and SC dynamic bus (secondary) **32**. A control card (CC) **36** is connected to LCs **24** via serial control bus **38**. The CC **36** is connected to SCs **24** via PCI bus **34**. A display card (DC) **42** may be connected to the CC **36** via DC buses **44**. One or more redundant cards may be provided for any of the cards(modules) described herein (plural SCs, LCs, CCs, Fcs may be provided). Also, Multiple PCI buses may be provided for redundancy. The architecture of the PGN **10** allows all major component types, making up the device **10**, to be identical. This allows for N+1 redundancy (N active components, 1 spare), or 1+1 redundancy (1 spare for each active component).

[**0036**] Several LCs **22** and several SCs **24** may be used as part of a single PGN **10**. The number may vary depending upon the access need (types of connection and number of users) as well as in dependance upon the redundancy provided. The LCs **22** each provide a network interface **11** for network traffic **13**. The LCs **22** handle all media access controller (MAC) and physical layer (Phy) functions for the system. The FC **20** handles inter-card routing of data packets. The SCs **24** each may implement forwarding path and protocol stacks.

[**0037**] The packets handled within the architecture are broadly categorized as fast path packets, special care packets, security packets and control packets. Fast path packets are those packets requiring protocol processing and protocol translation (converting from one protocol to another) at speeds greater than or equal to the rate at which they enter the device. Special care packets require additional processing in addition to the fast path packets. This might include Network Address Translation (NAT) or NAT coupled with application layer processing (NAT-ALG). Security packets require encryption, decryption authentication or the generation of authentication data. Control packets signal the start and end of data sessions, or are used to convey information to a particular protocol (i.e., the destination is unreachable). Control packets may also be dependent on interaction with external entities such as policy servers. The processing is divided according to the amount of processing required of the packet. The different classes of packet traffic are then dispatched to specialized processing elements so they may be processed concurrently. The concurrent nature of the processing allows for gains in throughput and speed not achievable by the usual serial processing approaches. In addition, all fast path processing is performed at a rate greater than or equal to that of the rate of ingress to the PGN **10**. This eliminates the need for any queuing of packets until the point at which they are awaiting transmission. Thus the users of the device do not experience delays due to fast path protocol processing or protocol translation.

[**0038**] Packet manipulation with respect to tunnel termination, encryption, queuing and scheduling takes place on the SC **24**. The master of the system is the CC **36**. The CC **36** manages the system, and acts as the point of communication with other entities in the network, i.e. the policy servers and the accounting manager.

[**0039**] The flexible routing therefore enables any service card **24** or line card **22**, in particular a spare service card **24** or line card **22**, to assume the role of another service card **24** or line card **22** by only changing the routing through the switch fabric card (FC) **20**. To support scalable performance, the PGN **10** divides the processing of in-bound protocols (e.g., the ingress path of LC1 **22'** through ingress processor **50** as shown in **FIG. 2B**), the out-bound protocols (e.g., the egress path of LC2 **22"** through egress processor **56** as shown in **FIG. 2B**) protocol control messaging, and the special handling of traffic requiring encryption.

[**0040**] Various protocols may be implemented. The Internet protocol (IP) preferably is used at the network layer functioning above the physical/link layer (physical infrastructure, link protocols—PPP, Ethernet, etc.) and below the application layer (interface with user, transport protocols etc.). The device **10** can be used with the IPSec protocol for securing a stream of IP packets. In such a situation, where secure virtual private networks are established the PGN **10** will perform ingress processing including implementing protocol stacks **55** in a software process including deencapsulating and deencrypting on the ingress side and implementing protocol stack **57** including encapsulating and encrypting on the egress side. **FIG. 4***a* illustrates this schematically with the ingress protocol stack **55** implementation being shown with processing proceeding from the IP layer **53** to the IP security layer **51**. This can involve for example deencapsulating and decrypting, protocol translating, authenticating, PPP terminating and NAT with the output being end-to-end packets. **FIG. 4***b* schematically illustrates the egress side protocol stack **57** implementation, wherein the end-to-end packets may be encapsulated, encrypted protocol translated, with authentication data generation, PPP generation and NAT. The IPSec encapsulation and/or encryption is shown moving from the IP security layer **51** to the IP layer **53**.

[**0041**] Any line card **22** can send traffic to any service card **24**. This routing can be configured statically or can be determined dynamically by the line card **22**. Any service card **24** can send traffic requiring ingress processing (e.g. from SC1 **24'** to SC2 **24"**) to any other service card **24** for ingress processing. Line cards **22** with the capability to classify ingress traffic can thus make use of unused capacity on the ingress service cards **24** by changing the routing.

[**0042**] Ingress processing **50** is physically separate from egress processing **56** (and also separate from processing at **52** and **54**). This enables ingress processing to proceed concurrently with egress processing resulting in a performance gain over a serialized approach. Any service card **24** handling ingress processing (e.g., at **50**) can send traffic to any other service card **24** for egress processing (e.g., at **56**). Thus, the device can make use of unused capacity that may exist on other service cards **24**.

[**0043**] The line cards (LC-x) **22** handle the physical interfaces. The line cards **22** are connected via the bus **38** to the (redundant) switch fabric card(s) (FC). Line card **22**s may be provided as two types, intelligent and non-intelligent. An intelligent line card **22** can perform packet classification (up to Layer **3**, network layer) whereas the non-intelligent line cards **22** cannot. In the former case, classified packets can be routed, via the FC **20**, to any service card **24** (SC) where ingress and egress processing occurs. This allows for load balancing since the LC **22** can route to the SC **24** with the least loaded ingress processor. In the latter case, the assignment of LCs **22** to SCs **24** is static, but programmable. Redundancy management is also made easier: In the event of failure of a line card **22**, a standby spare can be switched in by re-directing the flow through the FC **20**.

[**0044**] **FIG. 5** shows the arrangement of service cards **24** (SC-x). Each SC **24** provides ingress processing with ingress processing subsystem **62** (for fast path processing) and egress processing with physically separate egress processing subsystem **64** (for fast processing). The processing functions of these subsystems **62** and **64** are separate. Each ingress processing system contains separate paths **66** for special processing and separate components **68**, **70** and **73** for special processing. Each egress processing system contains a separate path **69** for special processing and the separate components **68**, **70** and **74** for special processing.

[**0045**] The role of the service cards, such as SC **24'**, is to process IP packets. IP packets enter the SC **24'** through the FC interface **20**; this is traffic coming, e.g., from LC1 **22'**.

[**0046**] Packets enter the ingress processor system **50**, where they are classified as subscriber data or control data packets. Control packets are sent up to one of two microprocessors, the control processor **70** or the special care processor **68**. Protocol stacks (e.g., **55** or **57**), implemented in software, process the packets at the control processor **70** or the special care processor **68**. A subscriber data packet is processed by the ingress processing subsystem **62** and or security subsystem **73** to produce an end-to-end packet (i.e. tunnels are terminated, IPSec packets are decrypted). The end-to-end packet is sent to another SC **24**" via the FC **20**. Packets enter the SC **24**" through the interface **72** to the FC **20**. The packets enter the egress processor system. This may be by use of another service card (e.g., SC **24**") where all the necessary encapsulation and encryption is performed. The packet is next sent to, e.g., LC**222**" that must transmit the packet into the network. Protocol stacks running on the control and special care processors may also inject a packet into the egress processor for transmission.

[**0047**] The flexible routing of ingress-to-egress, ingress-to-ingress (dividing ingress processing over more than one service card **24**) and egress-to-egress allows the device to dynamically adapt to changing network loads as sessions are established and torn down. Processing resources for ingress and egress can be allocated on different service cards **24** for a given subscriber's traffic to balance the processing load, thus providing a mechanism to maintain high levels of throughput. Typically, a subscriber data session is established on a given SC **24** for ingress and the same, or another SC **24** for egress. Information associated with this session, its context, is maintained or persists on the ingress and egress processor (e.g., of the processing subsystems **62** and **64**). The routing of ingress to ingress (e.g., from SC **24**' to SC **24**" via bus **32**, FC **20**, FC interface **72** and CSIX link **80**) permits the traffic to enter via a different LC **22** (because of the nature of the mobile user, such user could have moved and may now be coming in via a different path) and be handled by the ingress processing subsystem SC **24** holding the context (e.g., by Ingress processing subsystem **62** of SC **24**'). This eliminates the need to move the context at the price of maintaining context location. For example, the context information may be held and controlled by memory controller **76**. Moving context data can be problematic.

[**0048**] Processing subscriber data packets on the SC **24** occurs in one of three modes, fast path, security and special care path. Fast path processing is aptly named because it includes any processing of packets through the SC **24** at a rate greater than or equal to the ingress rate of the packets. These processing functions are implemented in the ingress processing subsystem **62** and egress processing subsystem **64** using custom-built hardware. Packets that require processing that cannot be done in the fast path are shunted off on the path **66** or **69** for either special care processing with processor **68** or security processing with processor **73** or **74**. Special care processing includes packets requiring PPP and GTP re-ordering or packets requiring NAT-ALG. Security processing is performed for IPSec packets or packets requiring IPSec treatment. When special care and security processing is completed, these packets are injected back into the fast path. Thus, while special care or security processing is in progress, the flow of packets not requiring such processing can proceed at a rate greater than or equal to their rate of the ingress. This method of concurrent processing eliminates the need to queue fast path packets thus enabling the device to sustain high and consistent levels of throughput.

[**0049**] The internal interfaces of PGN **10** enable the connections amongst ingress and egress processing functions. The ingress and egress PCI buses **66** and **69** are the central data plane interfaces from the control plane to the data plane. The ingress PCI bus **66** (see **FIG. 6**) provides a connection between the ingress processor field programable gate array (FPGA) **62**, encryption subsystem or security subsystem **73**, special care processor subsystem **68** and control processor subsystem **70**. The control processor subsystem **70** includes local system controller **86**, synchronous dynamic random access memory (SDRAM) **87**, cache **88**, global system controller **83** (providing a connection to PCI bus **34**), SDRAM **85** and control processor **90**. The global system controller **83**, the control processor **90** and the local system controller **86** are connected together via a bus connection **67**. The egress PCI bus **69** connects egress processor FPGA **81**, encryption subsystem or security subsystem **74**, special care processor **68** and control processor system **70**.

[**0050**] Each of the ingress PCI bus **66** and egress PCI bus **69** have an aggregate bandwidth of approximately 4Gb/s. They are used to pass data packets to and from the fast path hardware. For this reason, the egress processor FPGA **62** is the controller on the egress PCI bus **69**, and the ingress processor FPGA **64** (connected to egress processor **81**) is the controller on the ingress PCI bus **66**. These PCI buses **66** and **69** are shared with the control plane. Control plane functions on the PCI bus **34** are discussed below.

[**0051**] The special care subsystem **68**, the control processor system **70** and the security with switch fabric interface part (e.g., VSC872) **71**. Bus **91** carries data from the line card **22**' via bus **28** and via the FC **20**. The ingress processor subsystem **62** has a set of two (2) 3.2 Gb/s primary outputs with CSIX busses **77** with switch fabric interface part (e.g., VSC872) **72**" that will carry end to end data packets to the switch fabric (dynamic section) **20** for egress processing on the egress service card **24**". The connected service card (e.g., SC **24**") is packet dependent. The ingress processing element **62** has a secondary output in addition. This 3.2Gb/s bi-directional CSIX link **80**/**83** with switch fabric interface part (VSC872) **72**' to the switch fabric **20** is for ingress processor system **50** (e.g., of one SC **24**') to ingress processor **56** (cross service card, e.g., to another service card **24**") packet transfers.

[**0052**] The egress processing subsystem **64** receives data at inputs from two 3.2Gb/s CSIX links **77** out of the switch fabric interface part (e.g., VSC872) **72**". Packets coming to the egress processor subsystem **64** on these links have already been processed down to the end-to-end packet. The egress processor (e.g., **52** or **56**) sends a completely processed packet out to the line card **22** via a 3.2Gb/s CSIX link **95** to the switch fabric interface part **71**. The packet traverses the static switch fabric **20** on its way to the line card **22**.

[**0053**] The LC static buses **26**, and SC static buses **28**, interconnect line cards **22** and service cards **24** through the fabric card **20**. These connections are established when the control card configures the fabric card **20**. Connections made between LCs **22** and SCs **24** may be made to be virtually static. The connections may rarely change. Some reasons for connection changes are protection switchover and re-provisioning of hardware.

[**0054**] Each of the static buses **26** and **28** is comprised of 4 high-speed unidirectional differential pairs. Two pairs support subscriber data in the ingress direction while the other two pairs support subscriber data in the egress direction. Each differential pair is a 2.64384 Gbps high-speed LVDS channel. Each channel contains both clock and data information and is encoded to aid in clock recovery at the receiver. At this channel rate the information rate is 2.5 Gbps. Since unidirectional subscriber data flows in 2 channels, or pairs, between LCs **22** and SCs **24** for each static bus **26** and **28**, the aggregate information rate is 5 Gbps per direction per bus.

[**0055**] The primary dynamic buses **30** connect the ingress processor of one service card **24** to the egress processor of another service card **24** via the fabric card **20** on a frame-by-frame basis. Each primary dynamic bus **30** is comprised of 8 high-speed unidirectional differential pairs. Four pairs support subscriber data in the ingress direction while the other four pairs support subscriber data in the egress direction. Each differential pair is an 2.64384 Gbps high-speed LVDS channel. Each channel contains both clock and data information and is encoded to aid in clock recovery at the receiver. At this channel rate the information rate is 2.5 Gbps. Since unidirectional subscriber data flows in 4 channels, or pairs, the aggregate information rate for a given direction is 10 Gbps. Secondary dynamic buses **32** are electrically identical to the static buses but since they are dynamic, subscriber data may be rerouted on a frame-by-frame basis.

[**0056**] The process of the invention is illustrated generally in the flow diagram of **FIG. 8**. The process begins at **100** by providing the device infrastructure in the form of connection buses **28**, **30** and **32** and providing a switch fabric **20** for selectively interconnecting the connection buses. At least a first line card **22**', second line card **22**", a first service card **24**', a second service card **24**", and a control card **36** are provided. Advantageously a redundant line card **22**, redundant service card **24**, a redundant fabric card **20** and a redundant control card **36** may be provided. The fabric card **20** or fabric cards **20** are connected and configured to establish a substantially static connection from first line card **22**' via line card bus **26** through fabric card **20** to service card static bus **28** to service card **1** designated **24**'. In this configuration, the fabric card **20**, as indicated at **102**, also provides a connection from line card **22** designated **22**", the associated line card bus **26**, the fabric card **20** and the service card static bus **28** associated with service card **2** designated **24**". Step **104** shows the further steps of receiving packets at the first line card **22**' transferring the packets via LC bus **26**, fabric card **20**, SC static bus **28** to the first service card **24**'. As can be appreciated from **FIG. 5**, the first service card **24**' processes packets with ingress processing system **50**. As indicated above, control packets are sent to either control processor **62** or special care processor **66** and subscriber data packets are processed to produce the end-to-end packets as shown at **106**. At step **106** the necessary de-encapsulation and decryption are performed. As shown at **108**, the end-to-end packets are transferred via FC**20** to the egress processing system **56** of the second service card **24**" via dynamic bus **30** (primary dynamic bus). At step **110** the egress packet processor of second service card **24**" processes the end-to-end packets including encapsulation and encryption. The packets are then sent to a line card, such as second line card **22**" as indicated at step **112**. The line card then transmits packets into the network as shown at **114**. The protocol stack **55** running on the control processor **62** and special care subsystem **66** may also inject a packet into the ingress processor for transmission. The control processor **62** of service card **24**" and the special care processor **66** of service card **24**" may also treat further packets for egress processing

[**0057**] The entire system may be monitored using a display card **42** via display buses **44**. The line cards may be monitored via serial control buses **38**. The control card **36** may have other output interfaces such as EMS interfaces **48** which can include any one or several of 10/100 base T outputs **43** and serial output **47** and a PCMCIA (or compact flash) output **49**.

[**0058**] To support quality of service for multiple sets of customers, the device **10** supports a single point of queuing. Typically, a customer set **120**, each set **120** comprising multiple individuals, will be assured of a certain set of protocol services and a portion of the total bandwidth available within the device. It is therefore necessary to be able to monitor the rate of egress of the customer set's traffic. **FIG. 9** shows multiple customer sets **120** entering the device using different physical interfaces **22**.

[**0059**] Because of the distributed nature of the physical ingress, in particular because members of a customer set **120** may ingress on any physical interface and because all processing is performed at a rate greater than or equal to the ingress rate, a common point of aggregation is established on the egress portion of the SC. Referring to **FIG. 9**, customer set #**5** can enter the device using LC-**5** and LC-**7**. The ingress protocol processing for this customer set #**5** is hosted on SC-**3** and SC-**4** as indicated by ingress traffic **122** while egress processing is hosted on SC-**6** as shown by traffic after ingrees protocol processing **124**. The FC switches the ingress traffic from LC-**5** and LC-**7** to the two SCs **3** and **4** for ingress protocol processing. Since egress processing is hosted on SC-**6**, the FC **20** switches this traffic **124** to SC-**6** for egress processing following ingress protocol processing. SC-**6** provides the common point of aggregation and contains one or more queues (at the single location) for holding a customer set's traffic awaiting egress **126** to the LC. Queuing is necessary as the ingress rate of the customer set's aggregated traffic may, at times, exceed the egress rate of a particular physical interface. Monitoring of the egress rate of the customer set's traffic then occurs at the point of aggregation.

[**0060**] The invention provides a device based on modular units. The term card is used to denote such a modular unit. The modules may be added and subtracted and combined with identical redundant modules. However, the principals of this invention may be practiced with a single unit (without modules) or with features of modules described herein combined with other features in different functional groups.

[**0061**] While specific embodiments of the invention have been shown and described in detail to illustrate the application of the principles of the invention, it will be understood that the invention may be embodied otherwise without departing from such principles.

What is claimed is:

**1**. A network gateway device, comprising:

a physical interface for connection to a medium;

an ingress processor system for ingress processing of all or part of packets received from said physical interface and for sending ingress processed packets for egress processing;

an egress processor system for receiving ingress processed packets and for egress processing of all or part of received packets for sending to the physical interface;

interconnections including an interconnection between said ingress processor system and said egress processor system, an interconnection between said ingress processor system and said physical interface and an interconnection between said egress processor system and said physical interface.

**2**. A network gateway device according to claim 1, further comprising a packet queue establishing a queue of packets location awaiting transmission, said packet queue being the exclusive buffer location for packets between packets entering the device and packet transmission.

**3**. A network gateway device according to claim 1, wherein packets exit the device at a rate of the line established at the physical interface.

**4**. A network gateway device according to claim 1, wherein said ingress processing system processes packets including at least one or more of protocol translation, de-encapsulation, decryption, authentication, point-to-point protocol (PPP) termination and network address translation (NAT) and said egress processing system processes packets including at least one or more of protocol translation, encapsulation, encryption, generation of authentication data, PPP generation and NAT.

**5**. A network gateway device according to claim 1, wherein said ingress processor system includes a fast path processor subsystem processing packets at speeds greater than or equal to the rate at which they enter the device.

**6**. A network gateway device according to claim 5, wherein said fast path processor system provides protocol translation processing converting packets from one protocol to another protocol.

**7**. A network gateway device according to claim 5, wherein said egress processor system includes a fast path processor subsystem processing packets at speeds greater than or equal to the rate at which they are to leave the device.

**8**. A network device according to claim 5, wherein said ingress processor system includes a security processor subsystem for processing security packets requiring one or more of decryption and authentication, said processing occurring concurrently with fast path processor packet processing.

**9**. A network device according to claim 7, wherein said egress processor system includes a security processor subsystem for processing security packets requiring one or more of encryption and generation of authentication data, said processing occurring concurrently with fast path processor packet processing.

**10**. A network device according to claim 7, wherein said ingress processor system includes a special care packet processor for additional packet processing concurrently with fast path processor packet processing, said special care packet processor processing packets including one or more of network address translation (NAT) processing and NAT processing coupled with application layer processing (NAT-ALG).

**11**. A network device according to claim 7, wherein said ingress processor system includes a control packet processor for additional packet processing concurrently with fast path processor packet processing, including processing packets signaling the start and end of data sessions, packets used to convey information to a particular protocol and packets dependent on interaction with external entities.

**12**. A network device according to claim 1, wherein said physical interface includes a line card and said ingress processor system is provided as part of a service card and said egress processor system is provided in one of said service card and another service card and said interconnections include:

a line card bus connected to said line card;

a service card bus connected to at least one of said service card and said another service card; and

a switch fabric connecting said line card to at least one of said service card and said another service card.

**13**. A network device according to claim 12, wherein said service card includes said ingress processor system and said egress processor system and said another service card includes another ingress processor system for processing all or part of packets received from said line card and for sending ingress processed packets for egress processing and another egress processor system for receiving ingress processed packets and for processing all or part of received packets for sending to said line card, whereby packets may be sent between service cards for ingress processing by one service card and egress processing by another service card or for ingress processing using more than one service card.

**14**. A network gateway device according to claim 13, wherein each of said service cards is identical and a spare service cards is provided, for functionally replacing any one of the other service cards to provide redundancy.

**15**. A network gateway device according to claim 13, wherein said physical interface includes another line card connected by said switch fabric to at least one of said service card and said another service card.

**16**. A network gateway device according to claim 15, wherein said switch fabric connects any one of said line cards to any one of said service cards, whereby any line card can send packet traffic to any service card and routing of packet traffic is configured one of statically and dynamically by the said line card.

**17**. A network gateway device according to claim 13, wherein:

said service card bus includes a static bus part for connection of one of said service cards through said switch fabric to one of said line cards and a dynamic bus for connecting a service card to another service card through said fabric card allowing any service card to send packet traffic requiring ingress processing to any other service card for ingress processing and allowing any service card to send traffic requiring egress processing to any other service card for egress processing, whereby the system can make use of unused capacity that may exist on other service cards.

**18**. A network gateway device, comprising:

a plurality of line cards having a physical interface for connection to a medium and;

a plurality of service cards, each service card including an ingress processor for processing all or part of data received from one of said line cards and for sending ingress processed packets for egress processing and each of said service cards including an egress processor for receiving ingress processed packets and for processing all or part of received packets for sending to one of said line cards;

a line card bus connected to each of said line cards;

a service card bus connected to each of said service cards; and

a switch fabric connecting individual line cards to individual service cards, whereby packets may be sent between service cards for ingress processing by one service card and ingress processing by another service card or for shared ingress processing between more than one service card.

**19**. A network gateway device, comprising:

a first line card;

a first service card for packet processing including a first ingress processing system for at least one or more of de-encapsulation and decryption and a first egress processing system for at least one or more of encapsulation and encryption;

a second line card;

a second service card for packet processing including a second ingress processing system for at least one or more of de-encapsulation and decryption and a second egress processing system for at least one or more of encapsulation and encryption;

a switch fabric and connection interfaces connecting at least said first line card to said first service card, connecting said second line card to said second service card and connecting said first service card to said second service card.

**20**. A network system according to claim 19, wherein:

said connection interfaces include a static bus part for connection of one of said service cards through said switch fabric to one of said line cards and a dynamic bus for connecting a service card to another service card through said fabric card allowing any service card to send packet traffic requiring ingress processing to any other service card for ingress processing and allowing any service card to send traffic requiring egress processing to any other service card for egress processing, whereby the system can make use of unused capacity that may exist on other service cards.

**21**. A network system according to claim 19, wherein: each of said first ingress processing subsystem, said first egress processing subsystem, said second ingress processing subsystem and said second egress processing subsystem include physically separate packet processing.

**22**. A network gateway device according to claim 19, wherein each of said service cards is identical and a spare service cards is provided, for functionally replacing any one of the other service cards to provide redundancy.

**23**. A network gateway device according to claim 19, wherein said switch fabric connects any one of said line cards to any one of said service cards, whereby any line card can send packet traffic to any service card and routing of packet traffic is configured one of statically and dynamically to establish virtual traffic segregation for segregating traffic using one or more common service card and line card and to establish physical traffic segregation wherein traffic is segregated using groups of one or more service card and one or more line card.

**24**. A network gateway device according to claim 19, wherein said switch fabric connects any one of said line cards to any one of said service cards, whereby any line card can send packet traffic to any service card and routing of packet traffic is configured one of statically and dynamically by said line card.

**25**. A network gateway process, comprising:

receiving packets from a network via a physical interface connected to a medium;

ingress processing of packets, with an ingress processing system, including one or more of protocol translation processing, de-encapsulation, decryption, authentication, point-to-point protocol (PPP) termination and network address translation (NAT);

transferring packets to an egress packet processing subsystem;

egress processing said packets, with the egress processing system, including one or more of protocol translation, encapsulation, encryption, generation of authentication data, PPP generation and NAT processing.

**26**. A process according to claim 25, further comprising:

establishing a queue of packets awaiting transmission; and

transmitting queued packets via the physical interface, said packet queue being the exclusive buffer for packets between packets entering the ingress processing system and packet transmission.

**27**. A process according to claim 25, wherein packets are processed by said ingress processor at a rate of ingress at the physical interface.

**28**. A process according to claim 25, wherein said ingress processor system includes a fast path processor subsystem processing packets at speeds greater than or equal to the rate at which packets enter the ingress processor system.

**29**. A process according to claim 28, wherein said fast path processor system provides protocol translation processing converting packets from one protocol to another protocol.

**30**. A process according to claim 28, wherein said ingress processor system includes a security processor subsystem for processing security packets requiring one or more of decryption and authentication, said processing occurring concurrently with fast path processor packet processing.

**31**. A process according to claim 28, wherein said ingress processor system includes a special care packet processor for additional packet processing concurrently with fast path processor packet processing, said special care packet processor processing packets including one or more of network address translation (NAT) processing and NAT processing coupled with application layer processing (NAT-ALG).

**32**. A process according to claim 28, wherein said ingress processor system includes a control packet processor for additional packet processing concurrently with fast path processor packet processing, including processing packets signaling the start and end of data sessions, packets used to convey information to a particular protocol and packets dependent on interaction with external entities.

**33**. A process according to claim 28, further comprising:

providing said physical interface including a line card;

providing said ingress processor system as part of a service card;

providing said egress processor system is provided in one of the service card and another service,

providing a line card bus connected to the line card;

providing a service card bus connected to at least one of the service card and the another service card; and

providing a switch fabric connecting the line card to at least one of the service card and the another service card.

**34**. A process according to claim 25, further comprising:

providing said ingress processor system and said egress processor system as part of said service card;

providing another service card with another ingress processor system for processing all or part of packets received from said line card and for sending ingress processed packets for egress processing and another egress processor system for receiving ingress processed packets and for processing all or part of received packets for sending to the line card;

sending packets between service cards for ingress processing by one service card and egress processing by another service card or for ingress processing using more than one service card.

**35**. A process according to claim 33, further comprising:

providing another line card as part of said physical interface;

connecting said another line card, via said switch fabric to at least one of said service card and said another service card.

**36**. A process according to claim 35, further comprising:

using said switch fabric to connect any one of said line cards to any one of said service cards, whereby any line card can send packet traffic to any service card and routing of packet traffic is configured one of statically and dynamically by the said line card.

**37**. A process according to claim 33, further comprising:

providing said service card bus as a static bus for connection of one of said service cards through said switch fabric to one of said line cards and a dynamic bus for connecting a service card to another service card through said fabric card allowing any service card to send packet traffic requiring ingress processing to any other service card for ingress processing and allowing any service card to send traffic requiring egress processing to any other service card for egress processing, whereby the system can make use of unused capacity that may exist on other service cards.

**38**. A network gateway process according to claim 25, further comprising:

receiving packets from a network with a first packet protocol as part of said step of receiving packets;

using a first module ingress processing subsystem for said step of ingress processing of packets to produce end-to-end packets;

transferring the end-to-end packets to a second module egress packet processing subsystem;

using a second module egress processing subsystem for egress packet processing to produce packets for sending to a network with a second packet protocol;

receiving packets from the network with the second packet protocol;

using a second module ingress processing subsystem for ingress processing to produce end-to-end packets;

transferring the end-to-end packets to a first module egress processing subsystem;

using the first module egress packet processing subsystem for egress packet processing to produce packets for sending to the network with the first packet protocol.

**39**. The process according to claim 35, wherein

the first module is a service card for packet processing with the ingress processing subsystem separate from the egress processing subsystem and the second module is a service card for packet processing with the ingress processing subsystem separate from the egress processing subsystem.

**40**. The process according to claim 38, further comprising:

providing a switch fabric;

connecting a first line card to the switch fabric via a bus, the first line card providing a network interface;

connecting the first service card to the switch fabric via a bus;

connecting a second line card to the switch fabric via a bus, the second line card providing a network interface with the first packet protocol;

connecting the second service card to the switch fabric via a bus;

transferring packets from the first line card to the first service card via the fabric card and connected busses;

transferring packets from the first service card to the second service card via the fabric card and connected busses;

transferring packets from the second service card to the second line card via the fabric card and connected busses.

**41**. The process according to claim 40, further comprising:

transferring packets from the second line card to the second service card via the fabric card and connected busses;

transferring packets from the second service card to the first service card via the fabric card and connected busses;

transferring packets from the first service card to the first line card via the fabric card and connected busses.

**42**. A network gateway process according to claim 25, further comprising

providing a switch fabric;

connecting a first line card to the switch fabric via a bus, the first line card providing a network interface;

connecting a first service card to the switch fabric via a bus

connecting a second line card to the switch fabric via a bus, the second line card providing a network interface;

connecting a second service card to the switch fabric via a bus;

transferring packets from the first line card to the first service card;

processing packets at the first service card including one or more of de-encapsulation and decryption as part of said step of said step of ingress processing of packets;

transferring packets from the first service card to the second service card;

processing packets at the second service card including one or more of encapsulation and encryption as part of said step of egress processing packets;

transferring packets from the second service card to the second line card.

**43**. A process according to claim 42, wherein each of said first service card and said second service card process ingress packets from a line card, including encapsulation and encryption processing separate from processing egress packets to a line card, including de-encapsulation and decryption with separate processing subsystems.

**44**. The process according to claim 29, further comprising:

segregating traffic including physical segregating data traffic using one or more service card and one or more line card with traffic flows segregated from data traffic on one or more other service card and one or more other line card.

* * * * *

US 20050138202A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2005/0138202 A1**

**Seto** (43) **Pub. Date: Jun. 23, 2005**

(54) **ADDRESSES ASSIGNMENT FOR ADAPTOR INTERFACES**

(75) Inventor: **Pak-Lung Seto**, Shrewsbury, MA (US)

Correspondence Address:
**KONRAD RAYNES & VICTOR, LLP**
**Suite 210**
**315 S. Beverly Drive**
**Beverly Hills, CA 90212 (US)**

(73) Assignee: **Intel Corporation**

(21) Appl. No.: **10/742,302**

(22) Filed: **Dec. 18, 2003**

**Publication Classification**

(51) Int. Cl.$^7$ .................................................. **G06F 15/173**

(52) **U.S. Cl.** .............................................................. **709/238**

(57) **ABSTRACT**

Provided are a method and device for address assignment for adaptor interfaces. An initial configuration is maintained assigning multiple local interfaces to one initial local address. For each local interface, a remote address of a remote interface on at least one remote device to which the local interface connects is received. The initial local address is used to identify the local interfaces assigned to the initial local address in response to receiving a same remote address for each remote interface connected to the local interfaces assigned the initial local address.

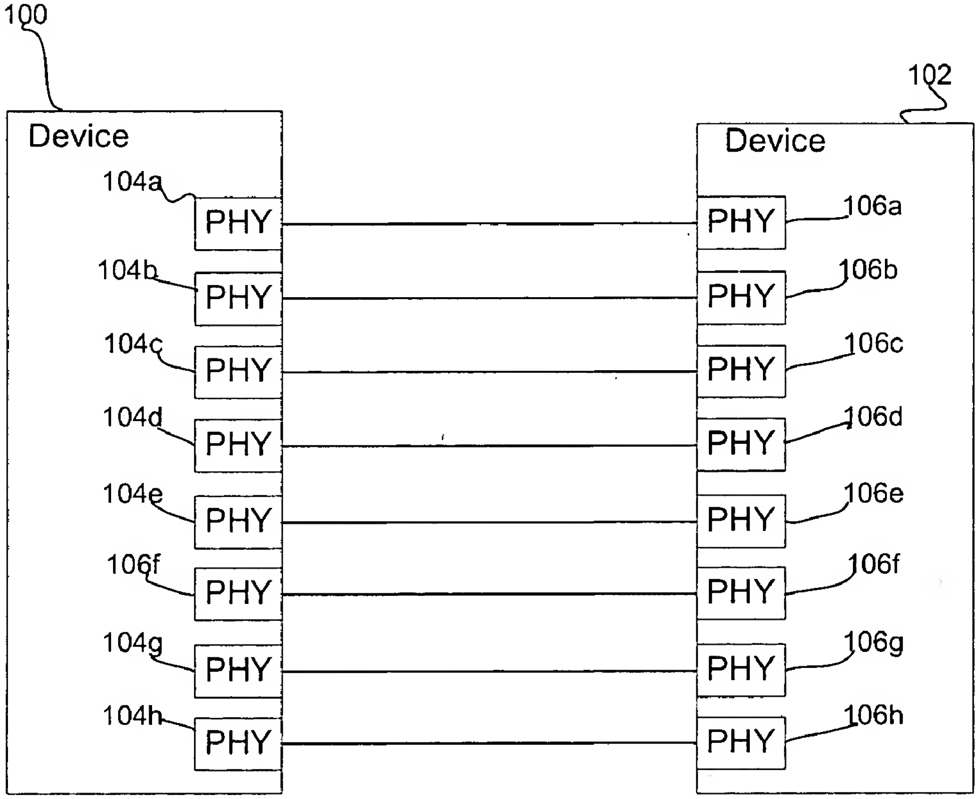


FIG. 1

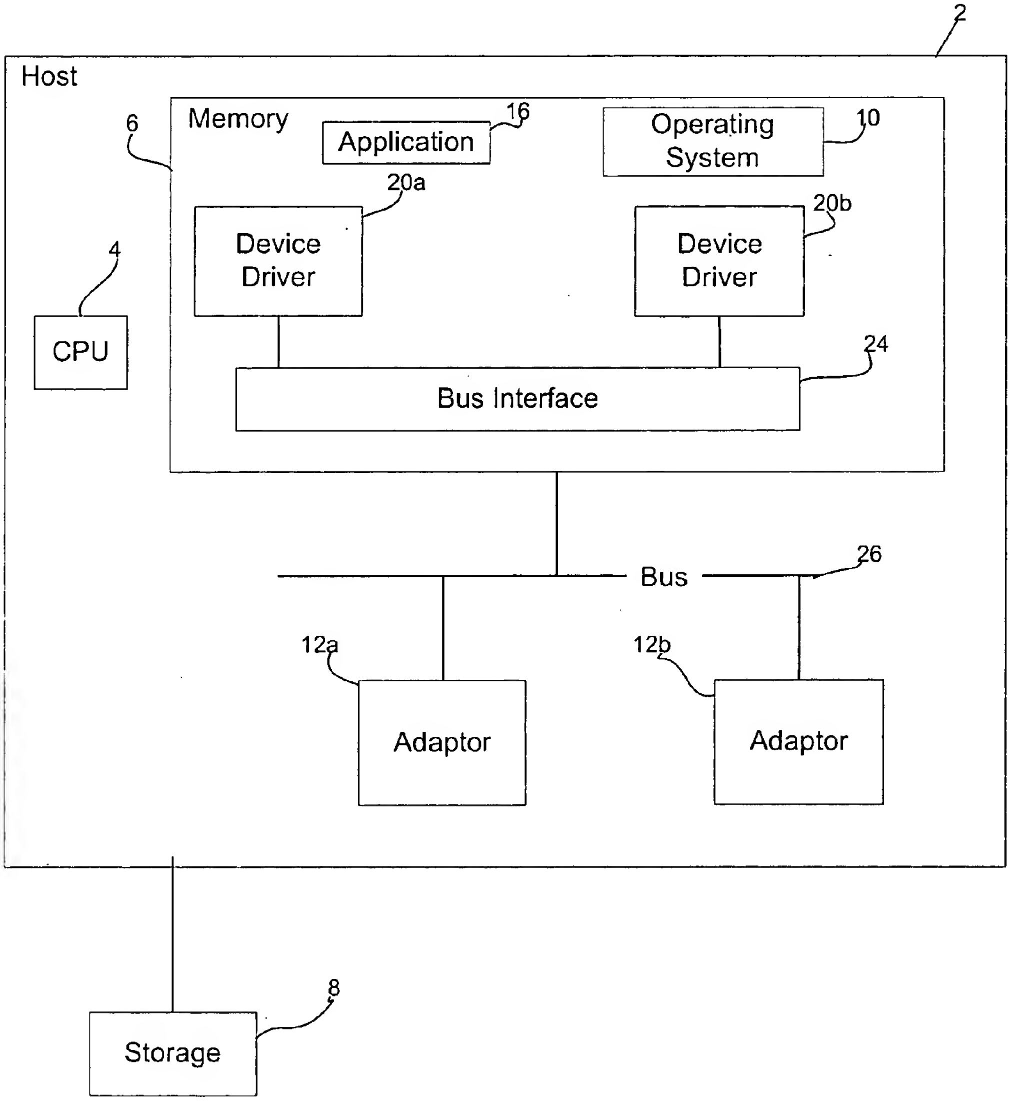

2
Host
6
Memory
Application
16
Operating System
10
20a
Device Driver
20b
Device Driver
4
CPU
24
Bus Interface
Bus
26
12a
12b
Adaptor
Adaptor
8
Storage

FIG. 2

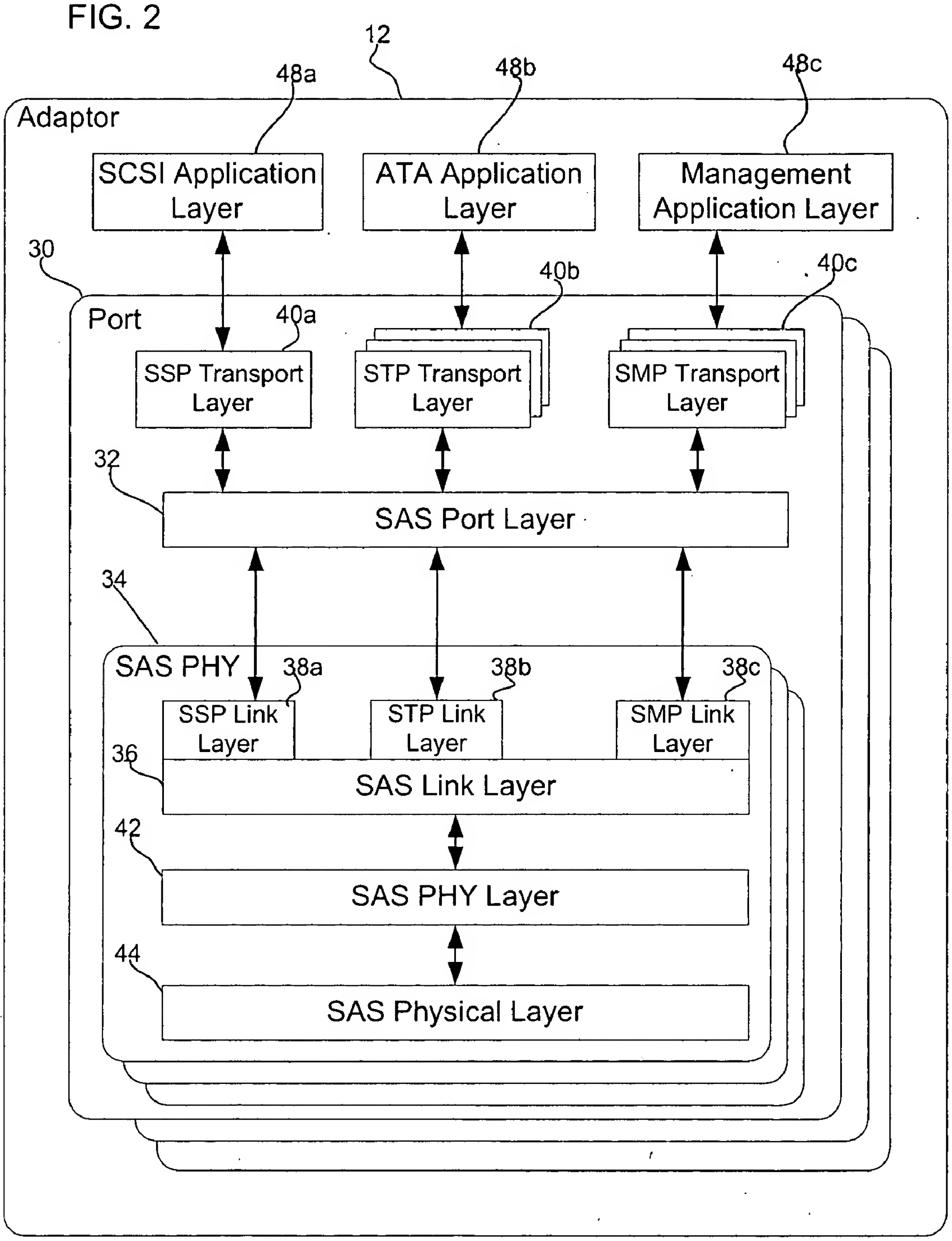
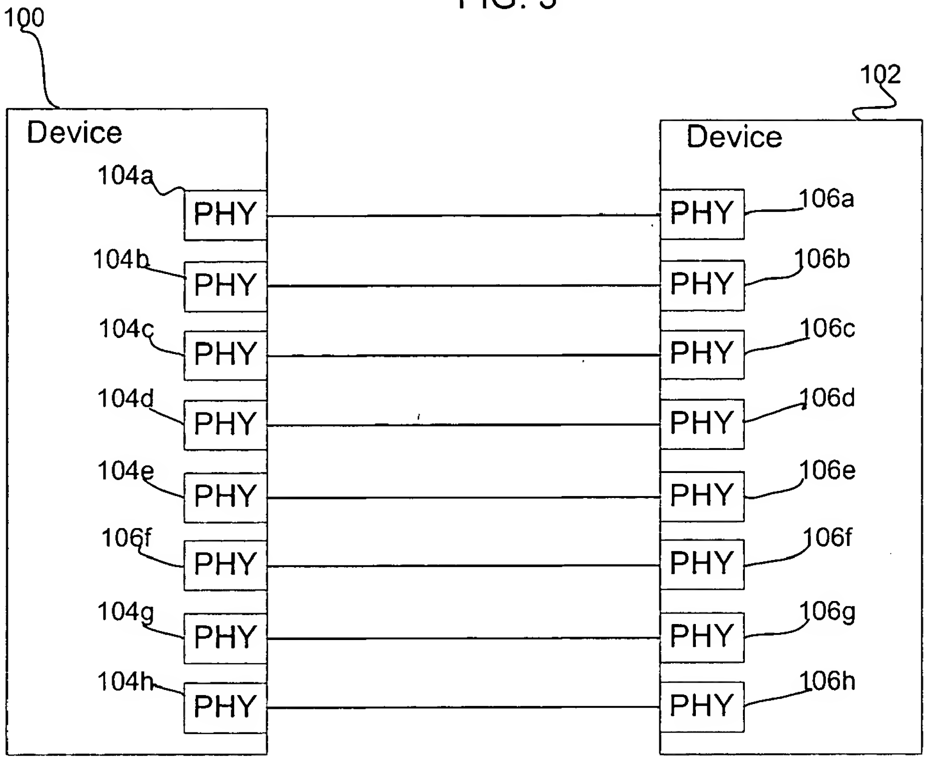

12
48a
48b
48c
Adaptor
SCSI Application Layer
ATA Application Layer
Management Application Layer
30
40b
40c
Port
40a
SSP Transport Layer
STP Transport Layer
SMP Transport Layer
32
SAS Port Layer
34
SAS PHY
38a
38b
38c
SSP Link Layer
STP Link Layer
SMP Link Layer
36
SAS Link Layer
42
SAS PHY Layer
44
SAS Physical Layer

FIG. 3


100
102
Device
Device
104a
104b
104c
104d
104e
106f
104g
104h
PHY
106a
106b
106c
106d
106e
106f
106g
106h

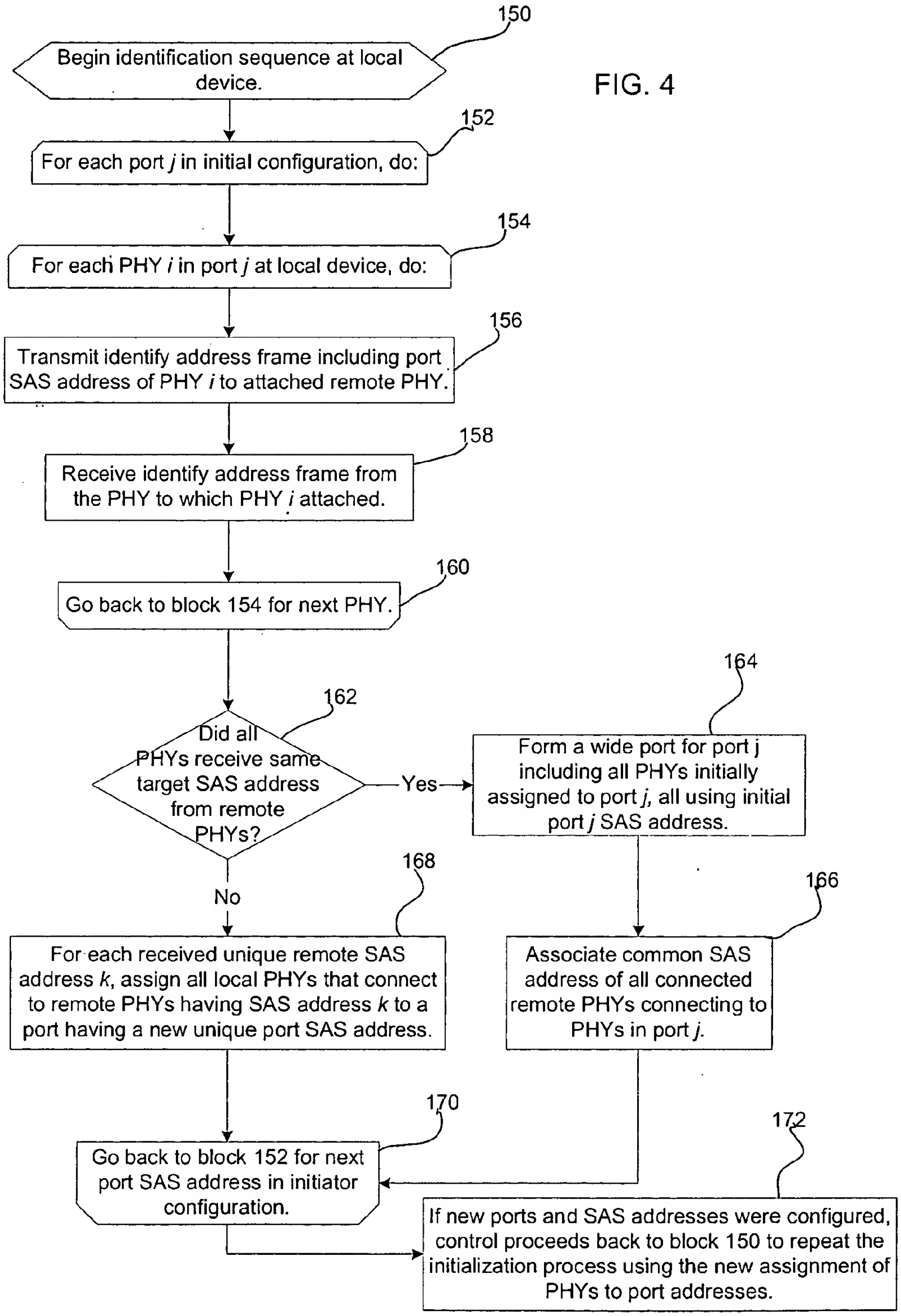

150
Begin identification sequence at local device.
152
For each port *j* in initial configuration, do:
154
For each PHY *i* in port *j* at local device, do:
156
Transmit identify address frame including port SAS address of PHY *i* to attached remote PHY.
158
Receive identify address frame from the PHY to which PHY *i* attached.
160
Go back to block 154 for next PHY.
162
Did all PHYs receive same target SAS address from remote PHYs?
Yes
No
164
Form a wide port for port j including all PHYs initially assigned to port *j*, all using initial port *j* SAS address.
166
Associate common SAS address of all connected remote PHYs connecting to PHYs in port *j*.
168
For each received unique remote SAS address *k*, assign all local PHYs that connect to remote PHYs having SAS address *k* to a port having a new unique port SAS address.
170
Go back to block 152 for next port SAS address in initiator configuration.
172
If new ports and SAS addresses were configured, control proceeds back to block 150 to repeat the initialization process using the new assignment of PHYs to port addresses.

FIG. 4

FIG. 5a

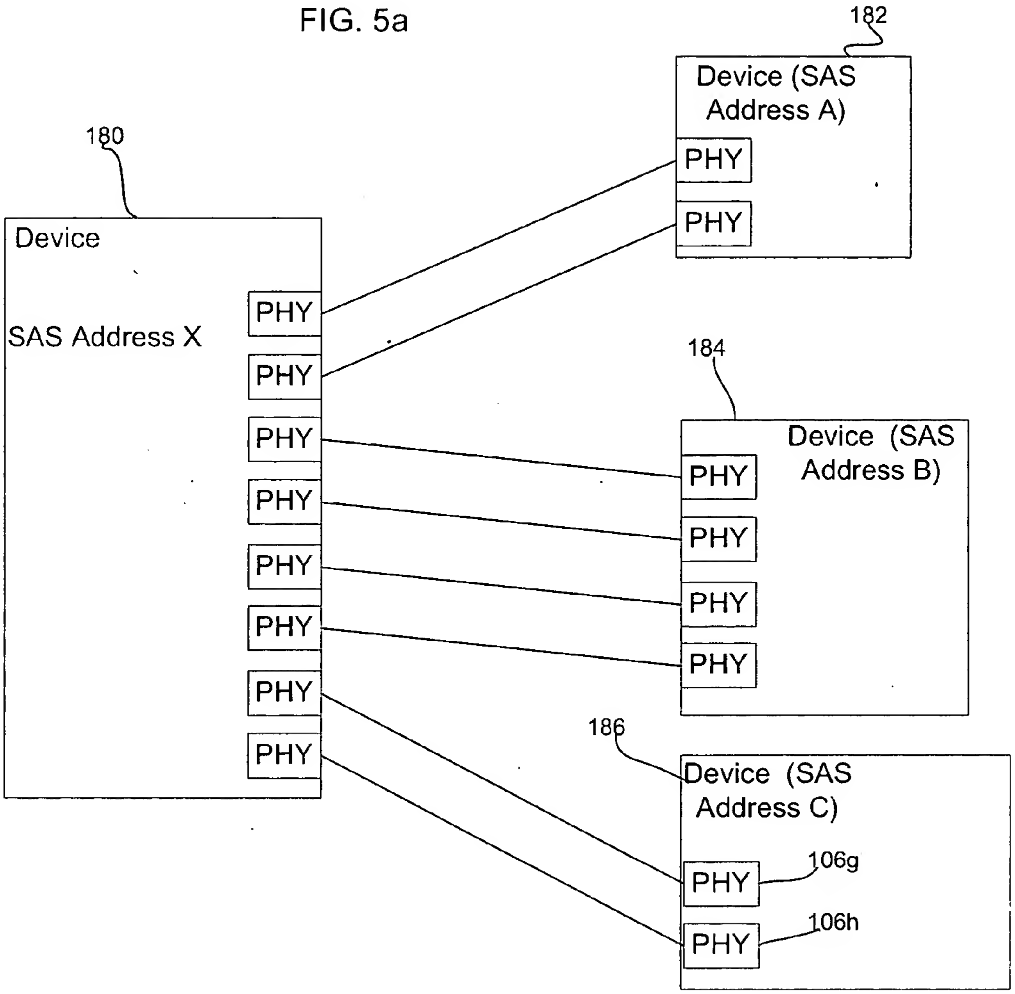
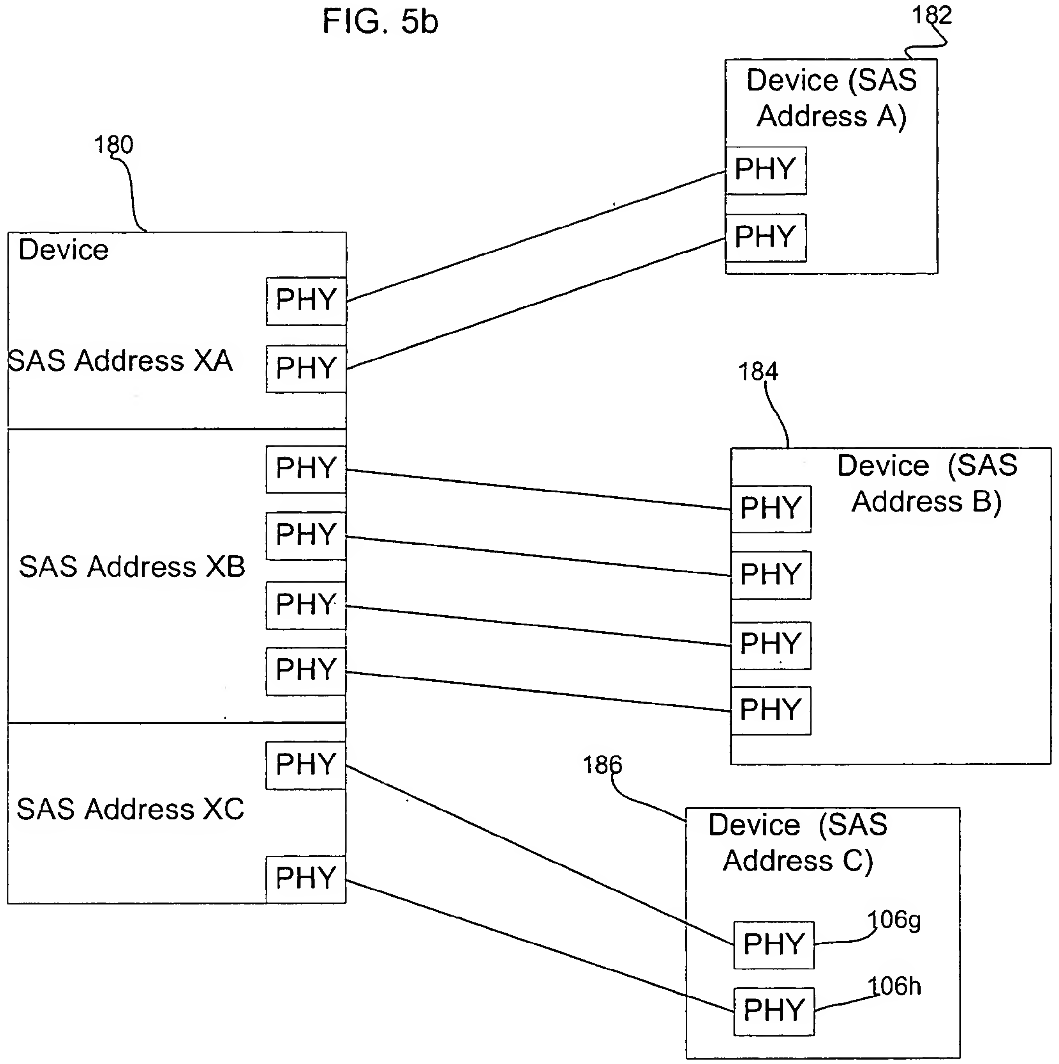

182
Device (SAS Address A)
PHY
PHY
180
Device
SAS Address X
PHY
PHY
PHY
PHY
PHY
PHY
PHY
PHY
184
Device (SAS Address B)
PHY
PHY
PHY
PHY
186
Device (SAS Address C)
PHY
106g
PHY
106h

FIG. 5b


182
Device (SAS Address A)
PHY
PHY
180
Device
PHY
SAS Address XA
PHY
184
Device (SAS Address B)
PHY
PHY
SAS Address XB
PHY
PHY
PHY
PHY
PHY
PHY
PHY
SAS Address XC
186
Device (SAS Address C)
PHY
106g
PHY
106h
PHY

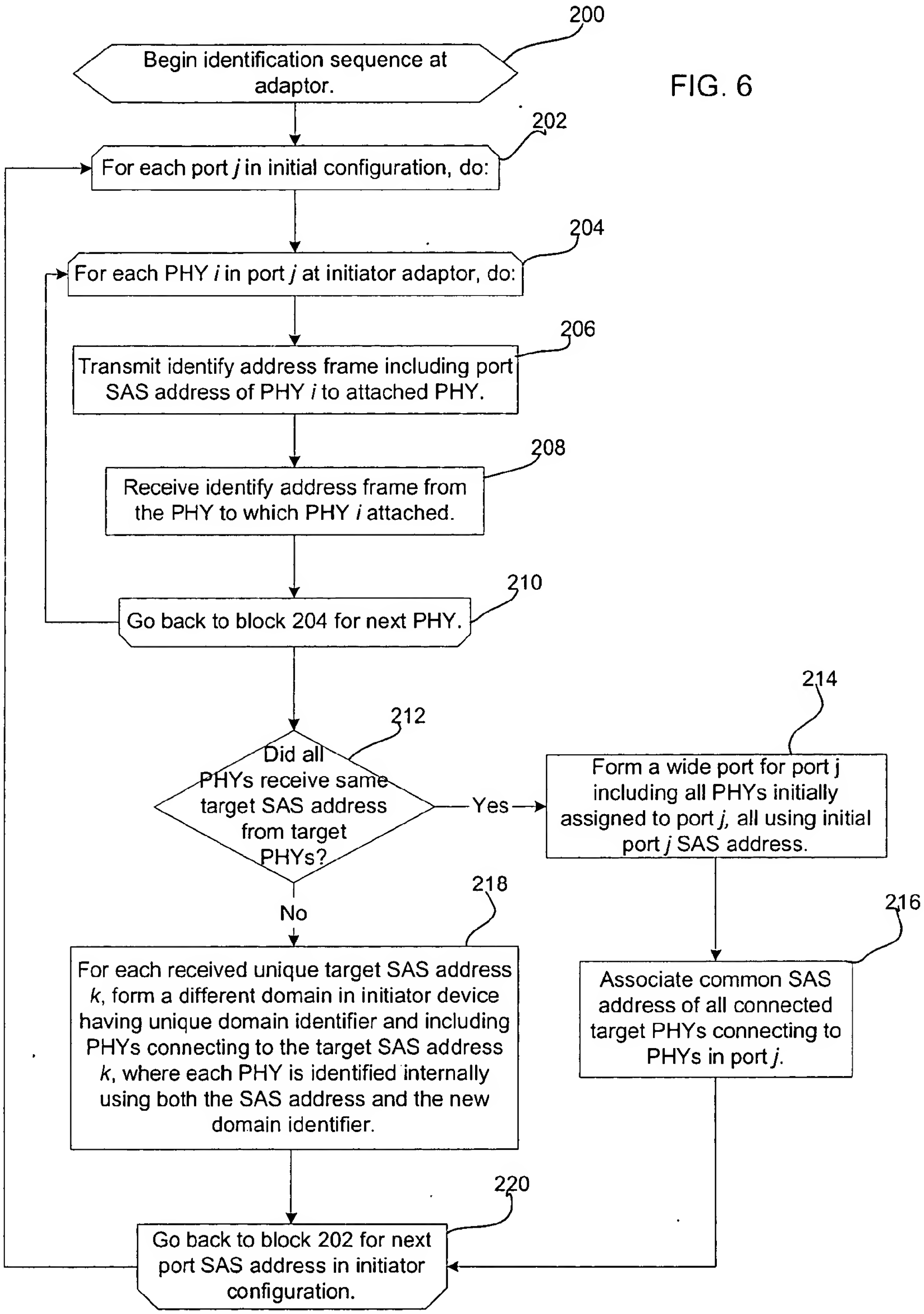
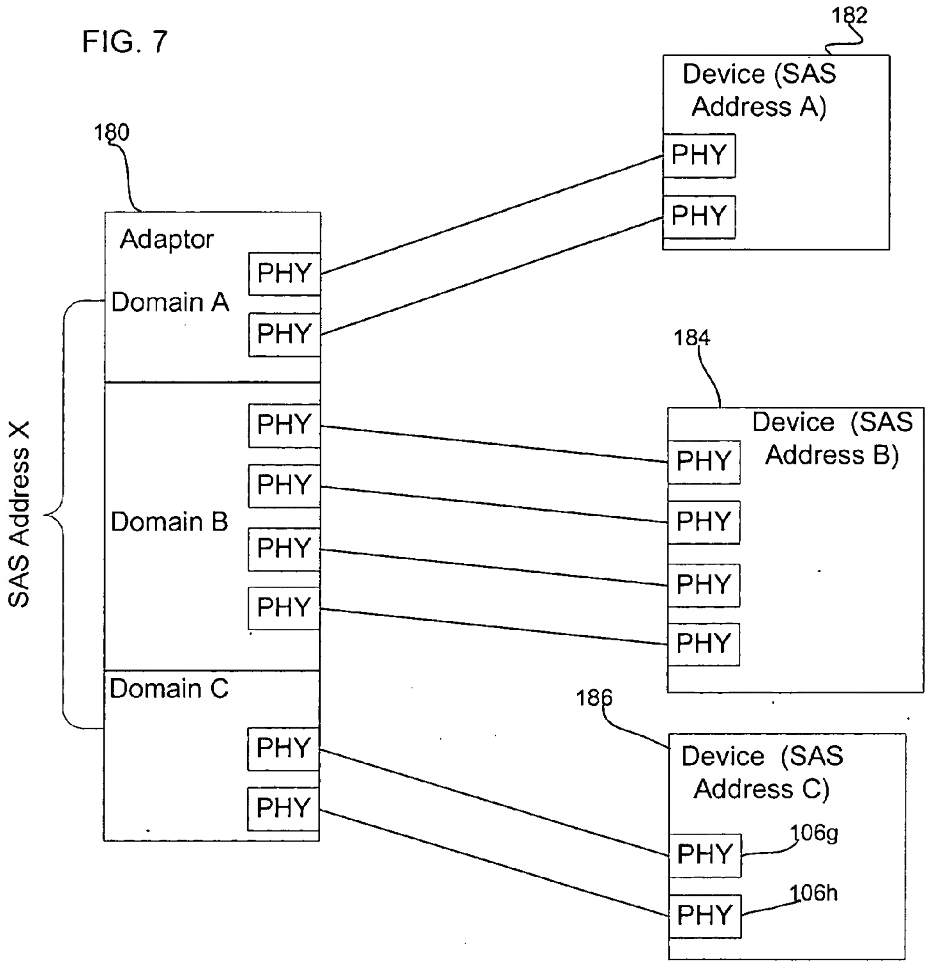

200
Begin identification sequence at adaptor.
202
For each port *j* in initial configuration, do:
204
For each PHY *i* in port *j* at initiator adaptor, do:
206
Transmit identify address frame including port SAS address of PHY *i* to attached PHY.
208
Receive identify address frame from the PHY to which PHY *i* attached.
210
Go back to block 204 for next PHY.
212
Did all PHYs receive same target SAS address from target PHYs?
Yes
No
214
Form a wide port for port j including all PHYs initially assigned to port *j*, all using initial port *j* SAS address.
216
Associate common SAS address of all connected target PHYs connecting to PHYs in port *j*.
218
For each received unique target SAS address *k*, form a different domain in initiator device having unique domain identifier and including PHYs connecting to the target SAS address *k*, where each PHY is identified internally using both the SAS address and the new domain identifier.
220
Go back to block 202 for next port SAS address in initiator configuration.

FIG. 6

FIG. 7


182
Device (SAS Address A)
PHY
PHY
180
Adaptor
Domain A
PHY
PHY
SAS Address X
Domain B
PHY
PHY
PHY
PHY
Domain C
PHY
PHY
184
Device (SAS Address B)
PHY
PHY
PHY
PHY
186
Device (SAS Address C)
PHY
106g
PHY
106h

# ADDRESSES ASSIGNMENT FOR ADAPTOR INTERFACES

## BACKGROUND

[0001] 1. Field

[0002] The embodiments relate to addresses assignment for adaptor interfaces.

[0003] 2. Description of the Related Art

[0004] An adaptor or multi-channel protocol controller enables a device coupled to the adaptor to communicate with one or more connected end devices over a physical cable or line according to a storage interconnect architecture, also known as a hardware interface, where a storage interconnect architecture defines a standard way to communicate and recognize such communications, such as Serial Attached Small Computer System Interface (SCSI) (SAS), Serial Advanced Technology Attachment (SATA), etc. These storage interconnect architectures allow a device to maintain one or more connections, such as direct point-to-point connections with end devices or connections extending through one or more expanders. Devices may also interconnect through a switch, an expander, a Fibre Channel arbitrated loop, fabric, etc. In the SAS/SATA architecture, a SAS port is comprised of one or more SAS PHYs, where each SAS PHY interfaces a physical layer, i.e., the physical interface or connection, and a SAS link layer having multiple protocol link layer. Communications from the SAS PHYs in a port is processed by the transport layers for that port. There is one transport layer for each SAS port to interface with each type of application layer supported by the port. A "PHY" as defined in the SAS protocol is a device object that is used to interface to other devices and a physical interface. Further details on the SAS architecture for devices and expanders is described in the technology specification "Information Technology—Serial Attached SCSI (SAS)", reference no. ISO/IEC 14776-150:200x and ANSI INCITS.***:200x PHY layer (Jul. 9, 2003), published by ANSI; details on the Fibre Channel architecture are described in the technology specification "Fibre Channel Framing and Signaling Interface", document no. ISO/IEC AWI 14165-25; details on the SATA architecture are described in the technology specification "Serial ATA: High Speed Serialized AT Attachment" Rev. 1.0A (January 2003).

[0005] Within an adaptor, the PHY layer may include the parallel-to-serial converter to perform the serial to parallel conversion of data, so that parallel data is transmitted to layers above the PHY layer, and serial data is transmitted from the PHY layer through the physical interface to the PHY layer of a receiving device. In the SAS specification, there is one set of link layers for each SAS PHY layer, so that effectively each link layer protocol engine is coupled to a parallel-to-serial converter in the PHY layer. The physical interfaces for PHYs on different devices may connect through a cable or through a path etched on the circuit board to connect through a circuit board path.

[0006] As mentioned, a port contains one or more PHYs. Ports in a device are associated with physical PHYs based on the configuration that occurs during an identification sequence. A port is assigned one or more PHYs within a device for those PHYs within that device that are configured to use the same SAS address within a SAS domain during the identification sequence, where PHYs on a device having the same SAS address in one port connects to PHYs on a remote device that also use the same SAS address within a SAS domain. A wide port has multiple interfaces, or PHYs and a narrow port has only one PHY. A wide link comprises the set of physical links that connect the PHYs of a wide port to the corresponding PHYs in the corresponding remote wide port and a narrow link is the physical link that attaches a narrow port to a corresponding remote narrow port. Further details on the SAS architecture is described in the technology specification "Information Technology—Serial Attached SCSI (SAS)", reference no. ISO/IEC 14776-150:200x and ANSI INCITS.***:200x PHY layer (Jul. 9, 2003), published by ANSI.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

[0008] **FIGS. 1 and 2** illustrate a system and adaptor in accordance with embodiments;

[0009] **FIGS. 3**, **5***a*, **5***b*, and **7** illustrate how devices may connect in accordance with embodiments; and

[0010] **FIGS. 4 and 6** illustrate operations to perform an identification sequence between connected devices in accordance with embodiments.

## DETAILED DESCRIPTION

[0011] In the following description, reference is made to the accompanying drawings which form a part hereof and which illustrate several embodiments. It is understood that other embodiments may be utilized and structural and operational changes may be made.

[0012] **FIG. 1** illustrates a computing environment in which embodiments may be implemented. A host system **2** includes one or more central processing units (CPU) **4** (only one is shown), a volatile memory **6**, non-volatile storage **8**, an operating system **10**, and adaptors **12***a*, **12***b* which includes physical interfaces to connect with remote deices, comprising end devices, switches, expanders, storage devices, servers, etc. An application program **16** further executes in memory **6** and is capable of transmitting and receiving transmissions via one of the adaptors **12***a*, **12***b*. The host **2** may comprise any computing device known in the art, such as a mainframe, server, personal computer, workstation, laptop, handheld computer, telephony device, network appliance, virtualization device, storage controller, etc. Various CPUs **4** and operating system **10** known in the art may be used. Programs and data in memory **6** may be swapped into storage **8** as part of memory management operations.

[0013] The operating system **10** may load a device driver **20***a* and **20***b* for each storage interface supported in the adaptor **12** to enable communication with a device communicating using the same supported storage interface and also load a bus interface **24**, such as a Peripheral Component Interconnect (PCI) interface, to enable communication with a bus **26**. Further details of PCI interface are described in the publication "PCI Local Bus, Rev. 2.3", published by the PCI-SIG. The operating system **10** may load device drivers **20***a* and **20***b* supported by the adaptors **12***a*, **12***b* upon detecting the presence of the adaptors **12***a*, **12***b*, which may occur during initialization or dynamically. In the embodiment of **FIG. 1**, the operating system **10** loads three device drivers **20***a* and **20***b*. For instance, the device drivers **20***a* and **20***b* may support the SAS and SATA storage interfaces, i.e., interconnect architectures. Additional or fewer device drivers may be loaded based on the number of storage interfaces the adaptors **12***a* and **12***b* supports.

[**0014**] **FIG. 2** illustrates an embodiment of an adaptor **12**, which may comprise the adaptors **12***a*, **12***b*. Each adaptor includes one or more ports **30**, where each port **30** contains a port layer **32** that interfaces with one or more SAS PHYs **34**. Each PHY includes a SAS link layer **36** having one or more protocol link layers. **FIG. 2** shows three protocol link layers, including a Serial SCSI Protocol (SSP) link layer **38***a* to process SSP frames, a Serial Tunneling Protocol. (STP) layer **38***b*, a Serial Management Protocol (SMP) layer **38***c*, which in turn interface through port layer **32** with their respective transport layers, a SSP transport layer **40***a*, a STP transport layer **40***b*, and an SMP transport layer **40***c*. The layers may be implemented as program components executed from memory and/or implemented in hardware.

[**0015**] Each PHY **34** for port **30** further includes a SAS PHY layer **42** and a physical layer **44**. The physical layer **44** comprises the physical interface, including the transmitter and receiver circuitry, paths, and connectors. As shown, the physical layer **44** is coupled to the PHY layer **42**, where the PHY layer **42** provides for an encoding scheme, such as 8b10b to translate bits, and a clocking mechanism. The PHY layer **32***a*, **32***b* . . . **32***n* may include a serial-to-parallel converter to perform the serial-to-parallel conversion and a phased lock loop (PLL) to track the incoming data and provide the data clock of the incoming data to the serial-to-parallel converter to use when performing the conversion. Data is received at the adaptor **12** in a serial format, and is converted at the SAS PHY layer **32***a*, **32***b* . . . **32***n* to the parallel format for transmission within the adaptor **12**. The SAS PHY layer **42** further provides for error detection, bit shift and amplitude reduction, and the out-of-band (OOB) signaling to establish an operational link with another SAS PHY in another device, speed negotiation with the PHY in the external device transmitting data to adaptor **12**, etc.

[**0016**] In the embodiment of **FIG. 2**, there is one protocol transport layer **40***a*, **40***b*, and **40***c* to interface with each type of application layer **48***a*, **48***b*, **48***c* in the application layer **50**. The application layer **50** may be supported in the adaptor **12** or host system **2** and provides network services to the end users. For instance, the SSP transport layer **46***a* interfaces with a SCSI application layer **48***a*, the STP transport layer **46***c* interfaces with an Advanced Technology Attachment (ATA) application layer **48***b*, and the SMP transport layer **46***d* interfaces with a management application layer **48***c*. Further details on the operations of the physical layer, PHY layer, link layer, port layer, transport layer, and application layer and components implementing such layers described herein are found in the technology specification "Information Technology—Serial Attached SCSI (SAS)". Further details of the ATA technology are described in the publication "Information Technology-AT Attachment with Packet Interface-6 (ATA/ATAPI-6)", reference no. ANSI INCITS 361-2002 (September, 2002).

[**0017**] Each port **30** has a unique SAS address across adaptors **12** and each PHY **34** within the port has a unique identifier within the adaptor **12** for management functions and routing. An adaptor **12** may further have one or more unique domain addresses, where different ports in an adaptor **12** can be organized into different domains or devices. The SAS address of a PHY may comprise the SAS address of the port to which the PHY is assigned and that port SAS address is used to identify and address the PHY to external devices in a SAS domain.

[**0018**] **FIG. 3** illustrates an example of how devices **100** and **102** may interface, where the device **100** has eight PHYs **104***a*, **104***b* . . . **104***j* linked to eight PHYs **106***a*, **106***b* . . . **106***j*, respectively, at the device **104**. The devices **100** and **102** may comprise a host, expander, storage device, server, etc., where the devices may implement the architecture described with respect to **FIG. 2** These devices **100** and **102** may have an initial address configuration for their PHYs, where the PHYs may share the same port address and be in the same domain. The initial address configuration for the PHYs in a device is based on user configuration selections.

[**0019**] **FIG. 4** illustrates operations implemented in a device implementing the architecture of **FIG. 2**, such as adaptor **12** devices **100** and **102**, to perform the identification sequence and configure the PHYs within ports. During the identification sequence, a device is informed of the address of remote interfaces, e.g., remote PHYs, connected to the local interfaces, e.g., local PHYs, of the device. The identification sequence operations in **FIG. 4** may be programmed in the port layer **32** of the adaptor **12**, devices **100**, **102** or performed by a device driver **20***a* and **20***b* for the adaptor **12**. Upon commencing (at block **150**) the identification sequence after a reset or power-on sequence at a device, e.g., **100**, a loop is performed at block **152** through **170** for each port j provided in the initial or default configuration maintained at the device, e.g., **100**. For each initial port j a loop is performed at blocks **154** through **160** for each PHY i assigned to port j in the initial configuration. At block **156**, a device, e.g., **100**, transmits identify address information including the SAS address of PHY i, which is the SAS address of port j, to the attached PHY, e.g., **106***a*, **106***b* . . . **106***h* in remote device **102**. The PHY i further receives (at block **158**) the identify address information from the PHY to which PHY i is attached. Device **100** may receive the identification information from the remote device **102** before transmitting identification information, or vice versa. Identification for a PHY is complete when a PHY has transmitted and received identification information. Further, if the device **100** does not receive identification information for the attached device PHY, then a timeout may occur where the entire link initialization process is restarted. Control then proceeds back to block **154** to transmit and receive the identify address information for the next PHY.

[**0020**] After all the PHYs, e.g., **104***a*, **104***b* . . . **104***h*, have received the identify address information from the attached PHYs, e.g., **106***a*, **106***b* . . . **106***h*, a determination is made (at block **162**) whether all the PHYs, e.g., **104***a*, **104***b* . . . **104***h*, received the same SAS address from the PHYs to which they connect. If so, then a wide port is formed for port j including all the PHYs, e.g., **104***a*, **104***b* . . . **104***h*, initially assigned to port j, so that all are configured to use the initial port j SAS address. The common SAS address of all the remote PHYs, e.g., **106***a*, **106***b* . . . **106***h*, is then associated with the common port j SAS address of the local PHYs, e.g., **104***a*, **104***b* . . . **104***h*, to use during operations. If (at block

**162**) the SAS addresses of the remote PHYs **106***a*, **106***b* . . . **106***h* are not the same, then for each received unique remote SAS address k, the local PHYs, e.g., **104***a*, **104***b* . . . **104***h*, that connect to remote SAS address k are assigned (at block **168**) to a newly configured port having a new unique port SAS address. The new unique SAS addresses of the local PHYs may not be the same if the connected remote PHYs were in different remote devices. In certain embodiments, the new unique port SAS addresses may be different than the initial SAS address configured for the port or one port SAS address may be the same as the initial SAS address and the other additional new SAS addresses for the connections to different remote devices may be unique. From block **166** or **168**, control proceeds (at block **170**) back to block **152** to consider any further ports in the initial configuration. After considering all ports in the initial configuration, if (at block **172**) new ports and SAS addresses were configured, control proceeds back to block **150** to perform a second instance of the initialization process using the new assignment of PHYs to port addresses.

[**0021**] The local and remote PHYs comprise local and remote interfaces at the local and remote devices, respectively. An interface is a physical or logical component that is connected to another interface on the same or a different device. The term interface may include interfaces other than PHY interfaces. A wide port comprises a port assigned multiple interfaces, where one or more interfaces may be assigned to a port. A local address, such as the local SAS address, comprises an address or identifier assigned to one or more interfaces and a remote address, such as the remote SAS address, comprises an address or identifier assigned to one or more interfaces in a remote device that connects to another interface, such as one of the local interfaces.

[**0022**] With the operations of **FIG. 4**, the ports are configured to include the maximum number of PHYs in each new port, where the PHYs in each new port will connect to PHYs in the connected adaptor that have the same SAS address. Further, if the PHYs in an initial port configuration are not connected to PHYs having the same PHY address, then new ports are configured with new SAS addresses to provide new ports, so that the PHYs assigned to the new ports connect to PHYs in the connected adaptors having the same SAS address. Further, after the reconfiguration of the ports, the identification sequence is performed again to perform configuration using the new port configuration.

[**0023**] **FIG. 5***a* illustrates an embodiment where the PHYs in the device **180** are configured to have one SAS address "x", which connect to PHYs in three different devices **182**, **184**, and **186**, each having a different SAS address "A", "B", and "C". Performing the operations of **FIG. 4** within a device having the configuration of **FIG. 5***a* results in the configuration shown in **FIG. 5***b*, in which adaptor **180** is configured to use three SAS addresses XA, XB, and XC to communicate with the PHYS in devices **182**, **184**, and **186**. Each of the SAS addresses XA, XB, and XC may comprise the address of a different port.

[**0024**] **FIG. 6** illustrates an alternative embodiment of operations to perform the identification sequence and establish port configurations. **FIG. 6** includes many of the same operations of **FIG. 4**, with the following exceptions. After determining (at block **212**) that the connected PHYs do not return the same address for a port j, instead of configuring new ports with different SAS addresses as done in **FIG. 4**, at block **218**, for each received unique target SAS address k, a different domain is formed in the device **180** having a unique domain identifier. Each PHY is then internally identified using both the SAS address and the newly configured domain identifier. After the domain designation is made, the device, e.g., **100** (**FIG. 3**), does not perform the identification sequence again and instead uses the domain identifier and SAS address to distinguish PHYs having the same address that are connected to different devices. However, external devices **182**, **184**, **186** may use the same SAS address to address the local PHYs.

[**0025**] **FIG. 7** illustrates an embodiment resulting from performing the operations of **FIG. 6** in a device having the configuration shown in **FIG. 5***a*, in which the device, e.g., **100**, is configured to use the same SAS address "X" for PHYs connected to different devices **252**, **254**, and **256**, but where those PHYs connected to different addresses are configured in different domains A, B, C. Thus, the device **250** uses the combination of domain identifier and SAS address to distinguish its local PHYs. With the embodiment of **FIG. 6**, a second identification sequence is not performed, unlike the second identification sequence performed at block **172** in **FIG. 4**, because there is no alteration of the default port configuration. Instead, the same address "X" is used. Thus, the remote devices **182**, **184**, **186** (**FIG. 7**) use the same SAS address to address the different PHYs in device **180** and the device **180** uses the domain addresses A, B, C in combination with the port SAS address "X" to distinguish the local PHYs. devices.

[**0026**] The described embodiments provide techniques for assigning PHYs or interfaces to ports when the interfaces receive different SAS addresses from the attached PHYs. The embodiment of **FIG. 6** minimizes communication and coordination between the local and remote PHYs, because the initial address configuration is used for interfaces that receive different addresses from the attached device and the device internally distinguishes interfaces connected to different addresses by assigning the interfaces to different domains.

[**0027**] In certain embodiments, the configuration is performed to form ports having a maximum possible width, i.e., maximum number of PHYs/connections. Maximizing the number of PHYs in a port maximizes the throughput for a port. Further, maximizing PHYs maximizes the load balancing opportunities. Yet further, maximizing the number of PHYs and connections at a port increases the number of alternate paths to the port, which minimizes I/O latency. Still further, maximizing the number of PHYs at a port provides redundant connections to allow continued operations should one or more PHYs fail.

## Additional Embodiment Details

[**0028**] The described embodiments may be implemented as a method, apparatus or article of manufacture using programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The term "article of manufacture" and "circuitry" as used herein refers to a state machine, code or logic implemented in hardware logic (e.g., an integrated circuit chip, Programmable Gate Array (PGA), Application Specific Integrated Circuit (ASIC), etc.) or a computer readable medium, such as magnetic storage medium (e.g., hard disk drives, floppy disks, tape, etc.), optical storage (CD-ROMs, optical disks, etc.), volatile and non-volatile memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, DRAMs, SRAMs, firmware, programmable logic, etc.). Code in the computer readable medium is accessed and executed by a processor. When the code or logic is executed by a processor, the circuitry may include the medium including the code or logic as well as the processor that executes the code loaded from the medium. The code in which preferred embodiments are implemented may further be accessible through a transmission media or from a file server over a network. In such cases, the article of manufacture in which the code is implemented may comprise a transmission media, such as a network transmission line, wireless transmission media, signals propagating through space, radio waves, infrared signals, etc. Thus, the "article of manufacture" may comprise the medium in which the code is embodied. Additionally, the "article of manufacture" may comprise a combination of hardware and software components in which the code is embodied, processed, and executed. Of course, those skilled in the art will recognize that many modifications may be made to this configuration, and that the article of manufacture may comprise any information bearing medium known in the art. Additionally, the devices, adaptors, etc., may be implemented in one or more integrated circuits on the adaptor or on the motherboard.

**[0029]** In the described embodiments, a physical interface was represented by a PHY, providing an interface between the physical connection and other layers within the adaptor. In additional embodiments, the interface representing a physical connection may be implemented using constructs other than a PHY.

**[0030]** Described embodiments utilize the SAS architecture. In alternative embodiments, the described techniques for assigning physical connections to ports may apply to additional storage interfaces.

**[0031]** In the described embodiments, certain operations were described with respect to layers within the device/adaptor architectures. In alternative implementations, the functions described as performed by a certain layer may be performed in a different layer.

**[0032]** In the described embodiments, transmissions are received at a device from a remote device over a connection. In alternative embodiments, the transmitted and received information processed by the transport protocol layer or device driver may be received from a separate process executing in the same computer in which the device driver and transport protocol driver execute.

**[0033]** In certain embodiments, the device driver and network adaptor embodiments may be included in a computer system including a storage controller, such as a SCSI, Redundant Array of Independent Disk (RAID), etc., controller, that manages access to a non-volatile storage device, such as a magnetic disk drive, tape media, optical disk, etc. In alternative implementations, the network adaptor embodiments may be included in a system that does not include a storage controller, such as certain hubs and switches.

**[0034]** In described embodiments, the storage interfaces supported by the adaptors comprised SATA and SAS. In additional embodiments, other storage interfaces may be supported. Additionally, the adaptor was described as supporting certain transport protocols, e.g. SSP, STP, and SMP. In further implementations, the adaptor may support additional transport protocols used for transmissions with the supported storage interfaces. The supported storage interfaces may transmit data at the same link speeds or at different, non-overlapping link speeds. Further, the physical interfaces may have different physical configurations, i.e., the arrangement and number of pins and other physical interconnectors, when the different supported storage interconnect architectures use different physical configurations.

**[0035]** The illustrated operations of **FIGS. 4 and 6** show certain events occurring in a certain order. In alternative embodiments, certain operations may be performed in a different order, modified or removed. Moreover, operations may be added to the above described operations and still conform to the described embodiments. Further, operations described herein may occur sequentially or certain operations may be processed in parallel. Yet further, operations may be performed by a single processing unit or by distributed processing units.

**[0036]** The adaptors **12***a*, **12***b* may be implemented in a network card, such as a Peripheral Component Interconnect (PCI) card or some other I/O card, or on integrated circuit components mounted on a system motherboard or backplane.

**[0037]** The foregoing description of various embodiments has been presented for the purposes of illustration and description. Many modifications and variations are possible in light of the above teaching.

What is claimed is:

**1**. A method, comprising:

maintaining an initial configuration assigning multiple local interfaces to one initial local address;

for each local interface, receiving a remote address of a remote interface on at least one remote device to which the local interface connects; and

using the initial local address to identify the local interfaces assigned to the initial local address in response to receiving a same remote address for each remote interface connected to the local interfaces assigned the initial local address.

**2**. The method of claim 1, further comprising:

generating at least one identifier in response to receiving multiple remote addresses from the remote interfaces connected to the local interfaces; and

assigning different identifiers to the local interfaces previously assigned the initial local address in response to generating the at least one identifier.

**3**. The method of claim 2, wherein the initial local address comprises a port address of a port to which the local interfaces are assigned as part of the initial configuration.

**4**. The method of claim 3, wherein each generated identifier comprises an additional port address, further comprising:

configuring an additional port in the device for each generated additional port address; and

assigning local interfaces to the ports, including the additional port and port having the initial local address.

**5**. The method of claim 4, wherein the local interfaces assigned to one port connect to remote interfaces having a same remote address.

**6**. The method of claim 2, wherein the at least one received remote address is received as part of an identification sequence, further comprising:

transmitting the initial local address to the remote interfaces connected to the local interfaces.

**7**. The method of claim 6, wherein the identifiers assigned to the local interfaces, including the at least one generated identifier, comprise local addresses, further comprising:

initiating an additional identification sequence in response to generating the at least one local address; and

transmitting the local addresses identifying the local interfaces to the connected remote interfaces in response to the additional identification sequence.

**8**. The method of claim 1, wherein the at least one remote device and a local device including the local interfaces implement the SAS architecture, wherein the local and remote addresses comprise SAS addresses, and wherein the local and remote interfaces comprise PHYs.

**9**. The method of claim 1, wherein the remote interfaces having different remote addresses are on different remote devices.

**10**. The method of claim 2, wherein generating the at least one identifier comprises generating a different identifier for each received different remote address, wherein a combination of the identifiers and the initial local address are used to identify the local interfaces assigned.

**11**. The method of claim 10, wherein the plurality of identifiers comprise domains and wherein the initial local address comprises a port address of a port to which the local interfaces are assigned as part of the initial configuration, wherein the local interfaces remain assigned to the port having the initial local address.

**12**. The method of claim 10, wherein the remote interfaces having different remote addresses are on different remote devices, wherein the combination of each of the plurality of identifiers and the default local address identify the local interfaces within a local device and wherein the initial local address identifies the local interfaces within the remote devices.

**13**. The method of claim 10, wherein the plurality of identifiers comprise domains, further comprising:

for each received remote address, generating a different domain in a local device including the local interfaces connected to the remote interfaces having the remote addresses.

**14**. The method of claim 13, wherein the generated domains include one domain in the initial configuration.

**15**. A device in communication with a plurality of remote interfaces on at least one remote device, comprising:

a plurality of local interfaces;

an initial configuration assigning multiple local interfaces to one initial local address;

circuitry capable of causing operations, the operations comprising:

(i) for each local interface, receiving a remote address of one remote interface to which the local interface connects; and

(ii) using the initial local address to identify the local interfaces assigned to the initial local address in response to receiving a same remote address for each remote interface connected to the local interfaces assigned the initial local address.

**16**. The device of claim 15, wherein the operations further comprise:

generating at least one identifier in response to receiving multiple remote addresses from the remote interfaces connected to the local interfaces; and

assigning different identifiers to the local interfaces previously assigned the initial local address in response to generating the at least one identifier.

**17**. The device of claim 16, wherein the initial local address comprises a port address of a port to which the local interfaces are assigned as part of the initial configuration.

**18**. The device of claim 17, wherein each generated identifier comprises an additional port address, and wherein the operations further comprise:

configuring an additional port in the device for each generated additional port address; and

assigning local interfaces to the ports, including the additional port and port having the initial local address.

**19**. The device of claim 18, wherein the local interfaces assigned to one port connect to remote interfaces having a same remote address.

**20**. The device of claim 16, wherein the at least one received remote address is received as part of an identification sequence, wherein the operations further comprise:

transmitting the initial local address to the remote interfaces connected to the local interfaces.

**21**. The device of claim 16, wherein the identifiers assigned to the local interfaces, including the at least one generated identifier, comprise local addresses, wherein the operations further comprise:

initiating an additional identification sequence in response to generating the at least one local address; and

transmitting the local addresses identifying the local interfaces to the connected remote interfaces in response to the additional identification sequence.

**22**. The device of claim 15, wherein the at least one remote device and the device implement the SAS architecture, wherein the local and remote addresses comprise SAS addresses, and wherein the local and remote interfaces comprise PHYs.

**23**. The device of claim 15, wherein the remote interfaces having different remote addresses are on different remote devices.

**24**. The device of claim 16, wherein generating the at least one identifier comprises generating a different identifier for each received different remote address, wherein a combination of the identifiers and the initial local address are used to identify the local interfaces assigned.

**25**. The device of claim 24, wherein the plurality of identifiers comprise domains and wherein the initial local address comprises a port address of a port to which the local interfaces are assigned as part of the initial configuration, wherein the local interfaces remain assigned to the port having the initial local address.

**26**. The device of claim 24, wherein the remote interfaces having different remote addresses are on different remote devices, wherein the combination of each of the plurality of identifiers and the default local address identify the local interfaces within the local device and wherein the initial local address identifies the local interfaces within the remote devices.

**27**. The device of claim 24, wherein the plurality of identifiers comprise domains, wherein the code is executed to further perform:

for each received remote address, generating a different domain in the local device including the local interfaces connected to the remote interfaces having the remote addresses.

**28**. The device of claim 27, wherein the generated domains includes one domain in the initial configuration.

**29**. A system in communication with at least one remote device having a plurality of remote interfaces, comprising:

a circuit board;

an adaptor coupled to the circuit board, comprising:

(i) a plurality of local interfaces;

(ii) an initial configuration assigning multiple local interfaces to one initial local address;

(iii) circuitry capable of causing operations to be performed, the operations comprising:

(a) for each local interface, receiving a remote address of one remote interface to which the local interface connects; and

(b) using the initial local address to identify the local interfaces assigned to the initial local address in response to receiving a same remote address for each remote interface connected to the local interfaces assigned the initial local address.

**30**. The system of claim 29, wherein the operations further comprising:

generating at least one identifier in response to receiving multiple remote addresses from the remote interfaces connected to the local interfaces; and

assigning different identifiers to the local interfaces previously assigned the initial local address in response to generating the at least one identifier.

**31**. The server of claim 30, wherein the initial local address comprises a port address of a port to which the local interfaces are assigned as part of the initial configuration.

**32**. An article of manufacture for interfacing local interfaces in a local device with connected remote interfaces in at least one remote device, wherein the article of manufacture causes operations to be performed, the operations comprising:

maintaining an initial configuration assigning multiple local interfaces to one initial local address;

for each local interface, receiving a remote address of one remote interface to which the local interface connects; and

using the initial local address to identify the local interfaces assigned to the initial local address in response to receiving a same remote address for each remote interface connected to the local interfaces assigned the initial local address.

**33**. The article of manufacture of claim 32, wherein the operations further comprise:

generating at least one identifier in response to receiving multiple remote addresses from the remote interfaces connected to the local interfaces; and

assigning different identifiers to the local interfaces previously assigned the initial local address in response to generating the at least one identifier.

**34**. The article of manufacture of claim 33, wherein the initial local address comprises a port address of a port to which the local interfaces are assigned as part of the initial configuration.

**35**. The article of manufacture of claim 34, wherein each generated identifier comprises an additional port address, wherein the operations further comprise:

configuring an additional port in the device for each generated additional port address; and

assigning local interfaces to the ports, including the additional port and port having the initial local address.

**36**. The article of manufacture of claim 35, wherein the local interfaces assigned to one port connect to remote interfaces having a same remote address.

**37**. The article of manufacture of claim 33, wherein the at least one received remote address is received as part of an identification sequence, wherein the operations further comprise:

transmitting the initial local address to the remote interfaces connected to the local interfaces.

**38**. The article of manufacture of claim 37, wherein the identifiers assigned to the local interfaces, including the at least one generated identifier, comprise local addresses, wherein the operations further comprise:

initiating an additional identification sequence in response to generating the at least one local address; and

transmitting the local addresses identifying the local interfaces to the connected remote interfaces in response to the additional identification sequence.

**39**. The article of manufacture of claim 32, wherein the at least one remote device and a local device including the local interfaces implement the SAS architecture, wherein the local and remote addresses comprise SAS addresses, and wherein the local and remote interfaces comprise PHYs.

**40**. The article of manufacture of claim 32, wherein the remote interfaces having different remote addresses are on different remote devices.

**41**. The article of manufacture of claim 33, wherein generating the at least one identifier comprises generating a different identifier for each received different remote address, wherein a combination of the identifiers and the initial local address are used to identify the local interfaces assigned.

**42**. The article of manufacture of claim 41, wherein the plurality of identifiers comprise domains and wherein the initial local address comprises a port address of a port to which the local interfaces are assigned as part of the initial configuration, wherein the local interfaces remain assigned to the port having the initial local address.

**43**. The article of manufacture of claim 41, wherein the remote interfaces having different remote addresses are on different remote devices, wherein the combination of each of the plurality of identifiers and the default local address identify the local interfaces within a local device and wherein the initial local address identifies the local interfaces within the remote devices.

**44**. The article of manufacture of claim 41, wherein the plurality of identifiers comprise domains, wherein the operations further comprise:

for each received remote address, generating a different domain in a local device including the local interfaces connected to the remote interfaces having the remote addresses.

**45**. The article of manufacture of claim 44, wherein the generated domains include one domain in the initial configuration.

**46**. The article of manufacture of claim 32, wherein the article of manufacture stores instructions that when executed result in performance of the operations.

* * * * *

US 20020188786A1

(19) **United States**

(12) **Patent Application Publication**

**Barrow et al.**

(10) **Pub. No.: US 2002/0188786 A1**

(43) **Pub. Date: Dec. 12, 2002**

(54) **DATA STORAGE SYSTEM WITH INTEGRATED SWITCHING**

(76) Inventors: **Jonathan J. Barrow**, Franklin, MA (US); **Frederick M. Rymsha**, Harvard, MA (US)

Correspondence Address:
**Christopher K. Gagne, Esq.**
**EMC Corporation**
**Office of the General Counsel**
**35 Parkwood Drive**
**Hopkinton, MA 01748 (US)**

(21) Appl. No.: **09/877,810**

(22) Filed: **Jun. 7, 2001**

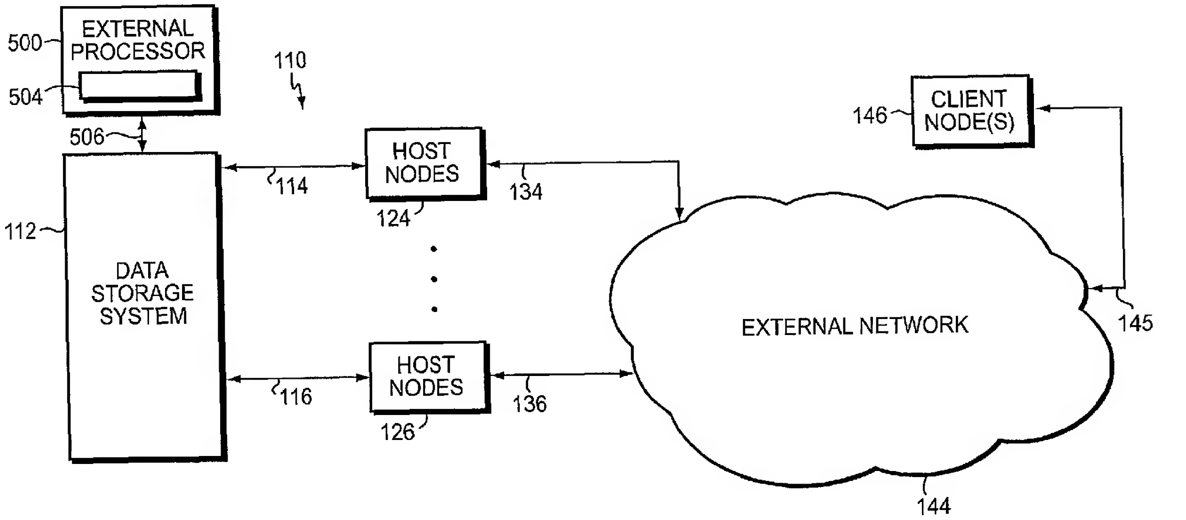
**Publication Classification**

(51) **Int. Cl.**$^7$ ............................ **G06F 13/00**; G06F 13/38

(52) **U.S. Cl.** ............................................................ **710/300**

(57) **ABSTRACT**

According to one embodiment of the present invention, a network adapter is provided that is used in the system to permit data communication among external data exchanging devices and an input/output (I/O) controller residing in the system. The adapter includes one or more interfaces that may be coupled to an electrical backplane in the system. The backplane is coupled to the controller, and is configured to permit communication between the controller and the adapter when the interfaces are coupled to the backplane. The adapter also includes an integrated switching system that has a first set of ports that may be coupled to the data exchanging devices and a second set of ports that may couple the switching system to the controller when the one or more interfaces are coupled to the backplane.



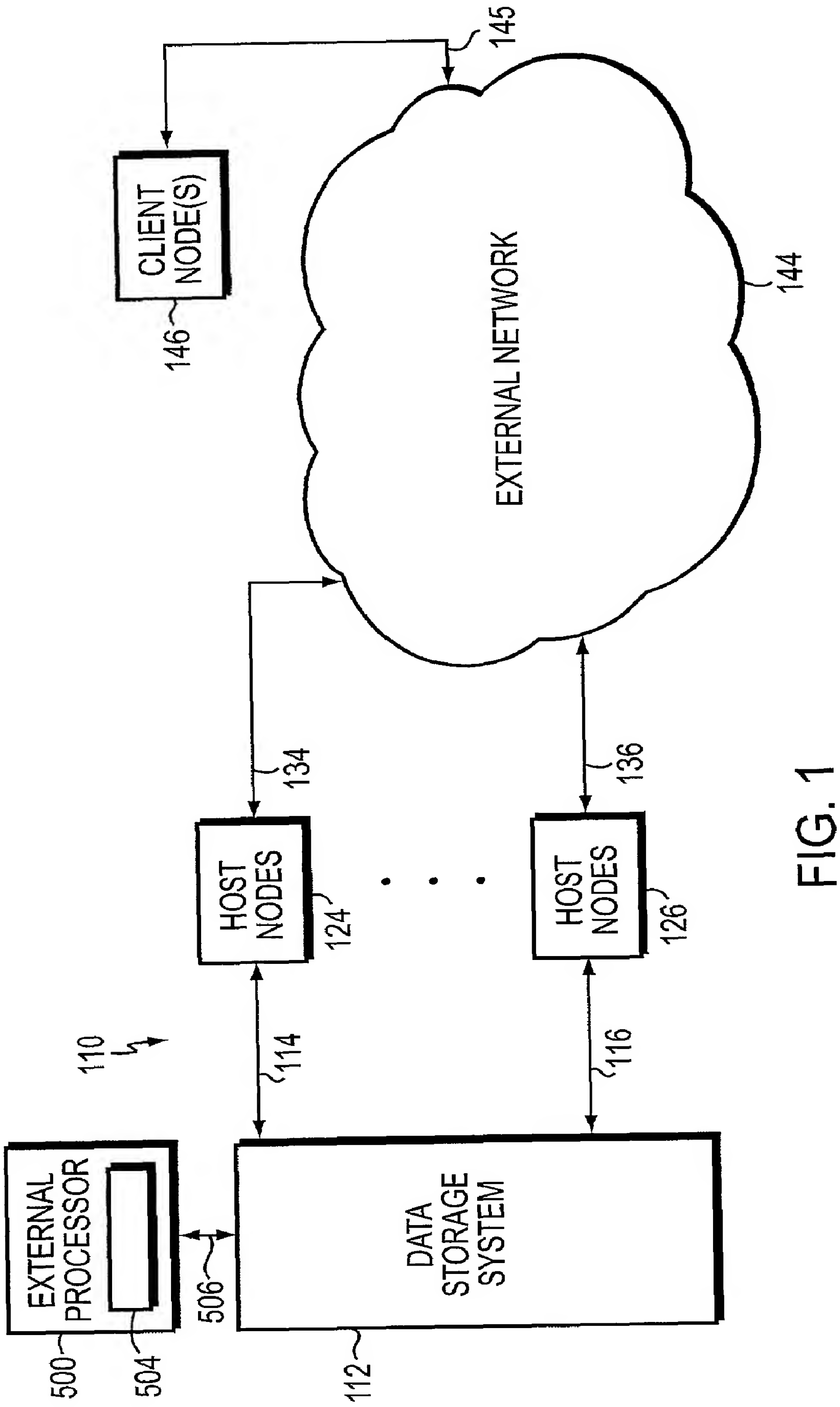

110
EXTERNAL PROCESSOR
500
504
506
DATA STORAGE SYSTEM
112
114
116
HOST NODES
124
HOST NODES
126
134
136
EXTERNAL NETWORK
144
145
CLIENT NODE(S)
146

FIG. 1

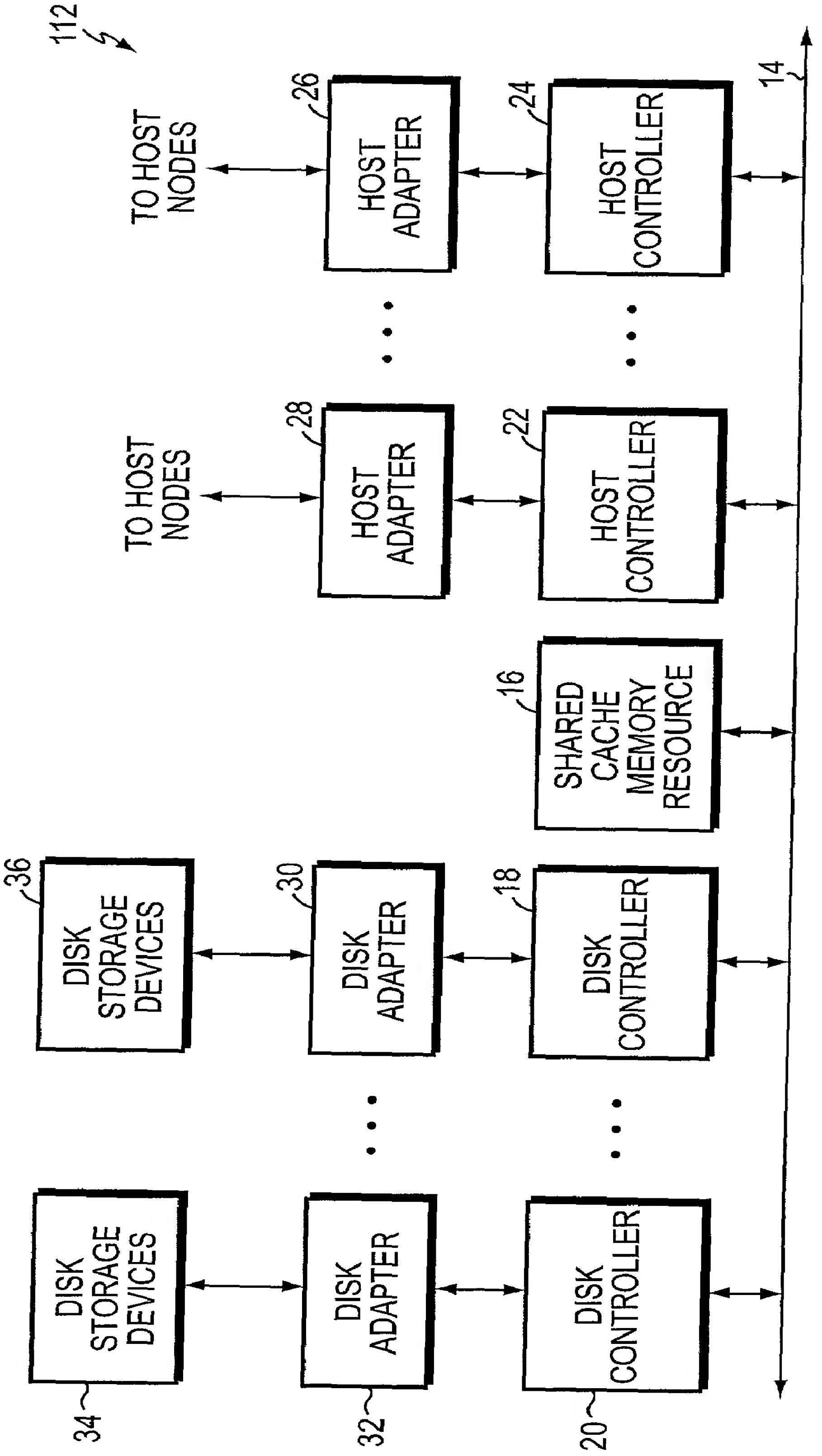

112
TO HOST NODES
26
HOST ADAPTER
24
HOST CONTROLLER
14
TO HOST NODES
28
HOST ADAPTER
22
HOST CONTROLLER
16
SHARED CACHE MEMORY RESOURCE
36
DISK STORAGE DEVICES
30
DISK ADAPTER
18
DISK CONTROLLER
DISK STORAGE DEVICES
34
DISK ADAPTER
32
DISK CONTROLLER
20

FIG. 2

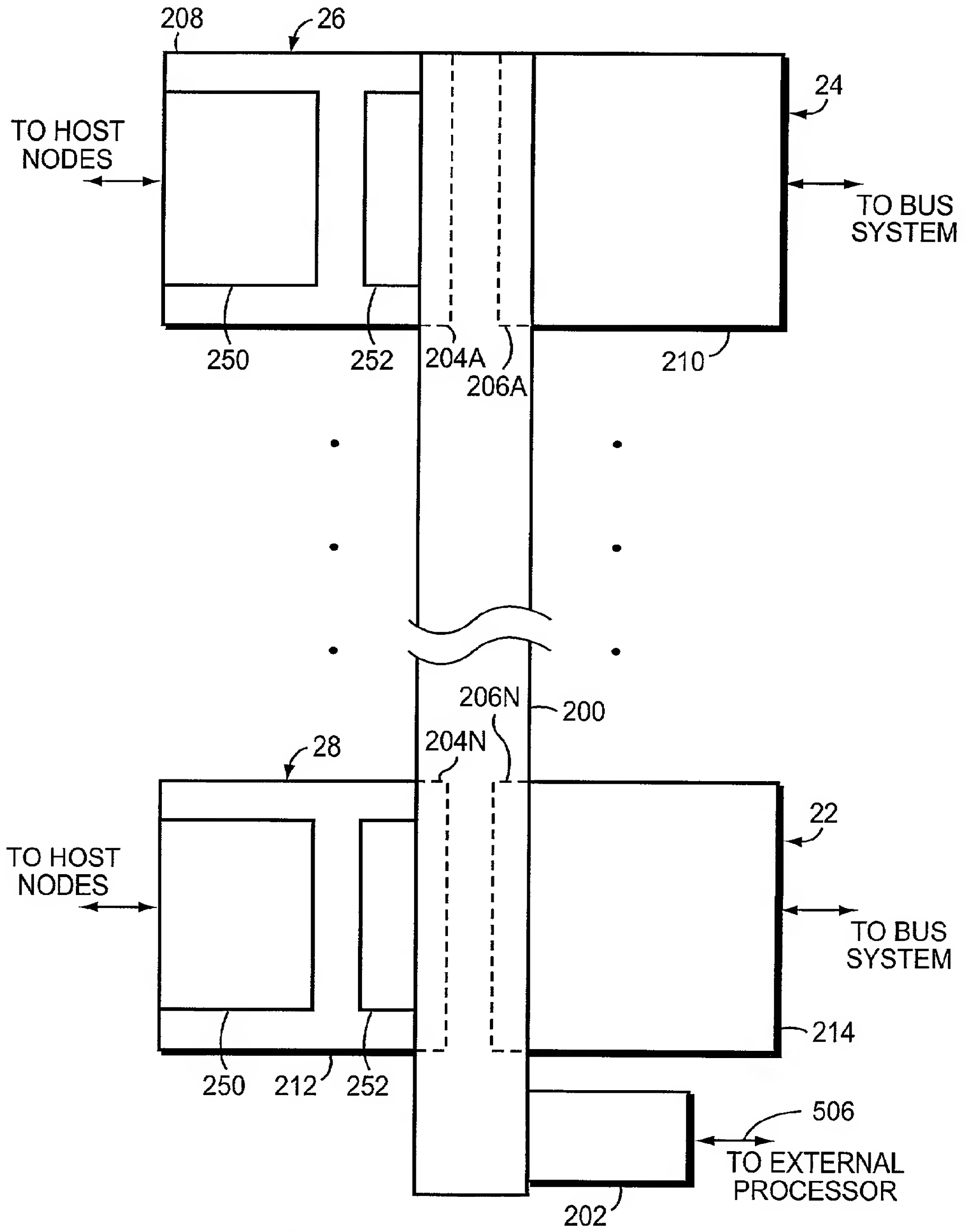

208
26
24
TO HOST NODES
TO BUS SYSTEM
250
252
204A
206A
210
206N
200
28
204N
22
TO HOST NODES
TO BUS SYSTEM
214
250
212
252
506
TO EXTERNAL PROCESSOR
202

FIG. 3

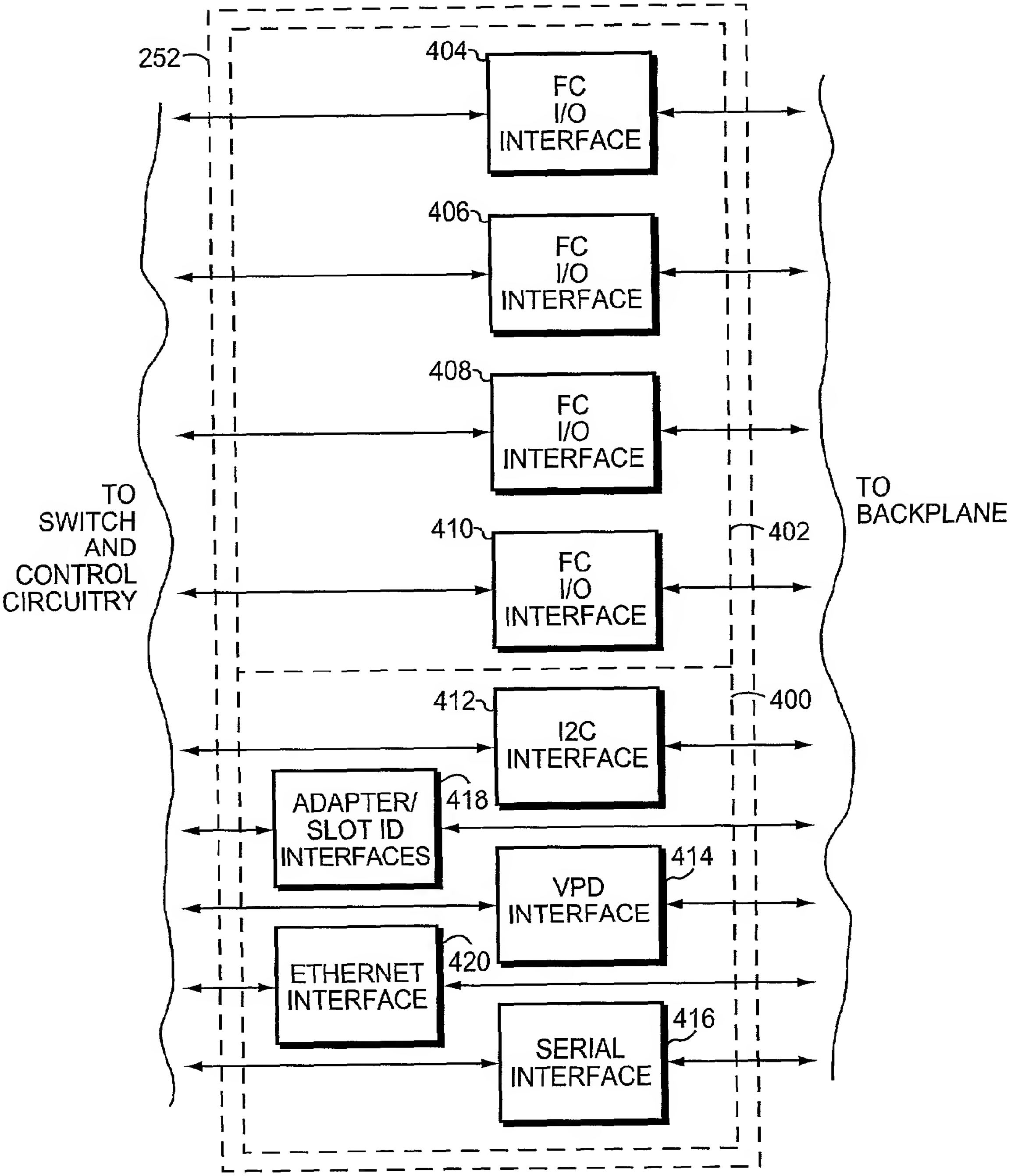

252
404
FC I/O INTERFACE
406
FC I/O INTERFACE
408
FC I/O INTERFACE
410
FC I/O INTERFACE
402
TO SWITCH AND CONTROL CIRCUITRY
TO BACKPLANE
412
I2C INTERFACE
400
ADAPTER/ SLOT ID INTERFACES
418
VPD INTERFACE
414
ETHERNET INTERFACE
420
SERIAL INTERFACE
416

FIG. 4

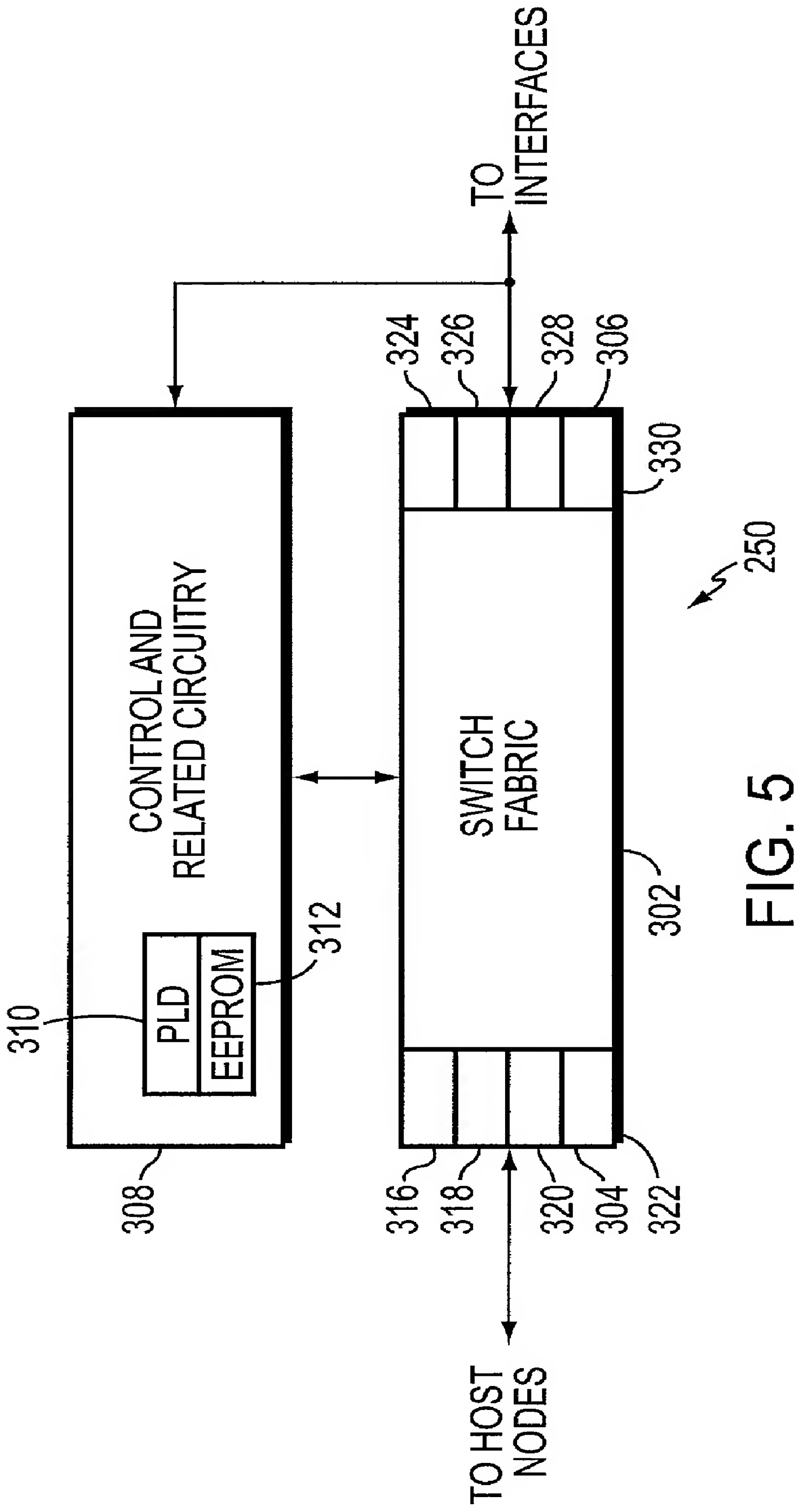

TO INTERFACES
CONTROL AND RELATED CIRCUITRY
PLD
EEPROM
310
312
308
SWITCH FABRIC
324
326
328
306
330
302
316
318
320
304
322
TO HOST NODES
250

FIG. 5

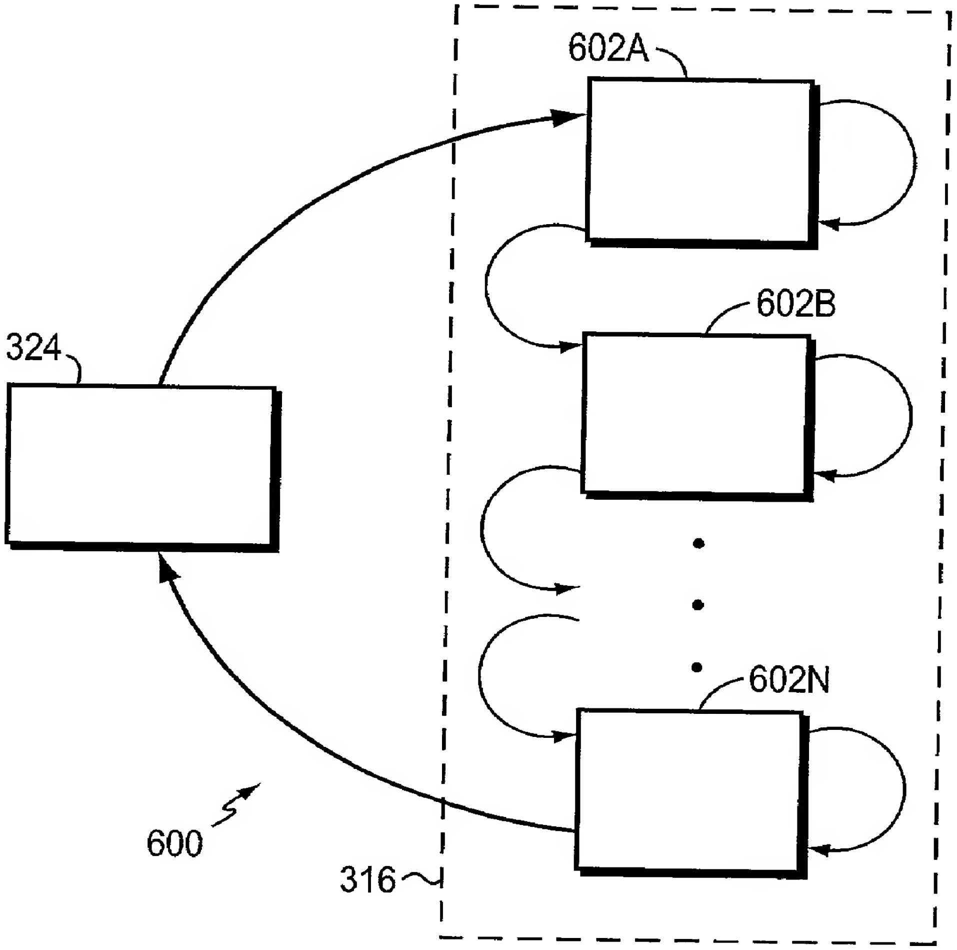


FIG. 6

# DATA STORAGE SYSTEM WITH INTEGRATED SWITCHING

## FIELD OF THE INVENTION

[0001] The present invention relates generally to a network adapter (and a method of using same) that may be used in a network data storage system to facilitate communication between the system and external data exchanging devices (e.g., host computer nodes), and more specifically, to such an adapter (and method of using same) wherein integrated switching capabilities may be used to facilitate data communication among the external data exchanging devices and an input/output (I/O) controller residing in the data storage system.

## BACKGROUND OF THE INVENTION

[0002] Network computer systems generally include a plurality of geographically separated or distributed computer nodes that are configured to communicate with each other via, and are interconnected by, one or more network communications media. One conventional type of network computer system includes a network data storage subsystem that is configured to provide a centralized location in the network at which to store, and from which to retrieve data. Advantageously, by using such a storage subsystem in the network, many of the network's data storage management and control functions may be centralized at the subsystem, instead of being distributed among the network nodes.

[0003] One type of conventional network data storage subsystem, manufactured and sold by the Assignee of the subject application (hereinafter "Assignee") under the tradename Symmetrix™ (hereinafter "the Assignee's conventional storage system"), includes a plurality of disk mass storage devices configured as one or more redundant arrays of independent (or inexpensive) disks (RAID). The disk devices are controlled by disk I/O controllers (commonly referred to as "back end" directors) that are coupled to a shared cache memory resource in the subsystem. The cache memory resource is also coupled to a plurality of host I/O controllers (commonly referred to as "front end" directors). The disk controllers are coupled to respective disk adapters that, among other things, interface the disk controllers to bus systems (e.g., small computer system interface (SCSI) based bus systems) used to couple the disk devices to the disk controllers. Similarly, the host controllers are coupled to respective host channel/network adapters that, among other things, interface the host controllers via channel input/output (I/O) ports to the network communications channels (e.g., Gigabit Ethernet, SCSI, Enterprise Systems Connection (ESCON), or Fibre Channel (FC) based communications channels) that couple the storage subsystem to computer nodes in the computer network external to the subsystem (commonly termed "host" computer nodes or "hosts").

[0004] In one conventional data storage network arrangement, a standalone network switch may be interjected in the communications channels intermediate to the host adapter I/O ports and the host nodes. More specifically, the host adapter channel I/O ports may be coupled to a first set of the switch's I/O ports, and a second set of the switch's I/O ports may be coupled to the host nodes. In this conventional data storage network arrangement, if the standalone network switch is appropriately configured, the host adapters (and their associated host controllers) may be able to exchange data/commands via the switch with any of the host nodes.

[0005] Unfortunately, standalone network switches tend to be relatively expensive and complex devices that may require substantial amounts of time and effort to install, configure, manage, and maintain in the data storage network. Also unfortunately, the presence of a standalone switch in the data storage network introduces into the network another stage, or hop, that the data must pass through when the data moves from the host nodes to the data storage system, and vice versa; this may increase latency in moving data from the host nodes to the data storage system, and vice versa.

## SUMMARY OF THE INVENTION

[0006] In accordance with the present invention, a network adapter and method of using same are provided that overcome the aforesaid and other disadvantages and drawbacks of the prior art. In one embodiment of the present invention, a network adapter is provided that is used in a network data storage system to facilitate data communication among external data exchanging devices and an I/O controller residing in the system. The data storage system may comprise a set of mass storage devices (e.g., disk mass storage devices) that may exchange data with the data exchanging devices via the adapter. The adapter may include one or more interfaces that may be physically coupled to a signal transmission medium/system (e.g., an electrical backplane) in the system. The backplane may be coupled to the controller, and may be configured to permit data communication between the controller and the adapter when the interfaces are coupled to the backplane. The adapter includes an integrated switching system (e.g., an FC switching fabric) that has a first set of ports that may be coupled to the data exchanging devices and a second set of ports that may couple the switching system to the controller when the one or more interfaces are coupled to the backplane.

[0007] The adapter may be embodied as an electrical circuit card that may be configured to be inserted into and received by a circuit card slot in the backplane in the data storage system. When the circuit card is inserted into and received by the slot, the card may be electrically and mechanically coupled to the backplane in the data storage system such that the one or more interfaces of the card are electrically coupled to the backplane.

[0008] The adapter may be assigned a first network layer address based at least partially upon a slot identification number that identifies the location of the backplane circuit card slot in which the adapter card is inserted and resides. The first network layer address may be changed during a configuration of the data storage system to a second network layer address.

[0009] The one or more interfaces of the adapter may comprise at least one interface through which a command may be issued to the adapter to cause the adapter to change from a first (operational) mode to a second (diagnostic) mode of operation. For example, the one or more interfaces of the adapter may comprise a first interface and a second interface. The first interface may permit a processor that is external to the adapter card, the controller, and the data exchanging devices to issue a management or diagnostic testing-related command to the adapter card via the backplane. Optionally, the external processor also may be external to the data storage system itself, may be coupled to the adapter via a network, and may access the adapter card via the network, using the second network layer address. The second interface may permit the controller to issue a management or diagnostic command to the adapter card via the backplane. The one or more interfaces of the adapter may also comprise a third interface that may permit configuration-related information (e.g., information related to the configuration of the adapter) to be retrieved via the backplane from a non-volatile memory (e.g., comprising one or more electrically erasable/programmable read only memory (EEPROM) devices) comprised in the adapter card.

[0010] In the diagnostic mode of operation, a diagnostic test of the adapter may be performed. The diagnostic test may comprise either (1) a built-in self-test (BIST) of the adapter or (2) a second, special type of test of the adapter that is different from the BIST of the adapter. This second type of test of the adapter may include transmission of a respective test vector along a first circuitous test path or loop in the adapter. The first test path may both begin and terminate at a first I/O port that couples the adapter to the controller when the adapter's interfaces are coupled to the backplane; the first test path may include a subset of the first set of ports of the switching system. This second type of test of the adapter may also include the transmission of a respective test vector along a second circuitous test path or loop in the adapter. The second test path may both begin and terminate at a second, different I/O port that may couple the adapter to the controller when the adapter's interfaces are coupled to the backplane. The second test path may include a second, different subset of the first set of ports of the switching system.

[0011] In summary, a network adapter according to the present invention includes an integrated switching system. The adapter may be configured for insertion into a network data storage system, and when inserted into the network data storage system, one or more interfaces comprised in the adapter may be coupled to a signal transmission medium in the data storage system. When the one or more interfaces are so coupled to the signal transmission medium, an I/O controller in the data storage system may be able to exchange data with the adapter via the medium, and the integrated switching system may be used facilitate communication among external data exchanging devices (e.g., host computer nodes) and the controller in the data storage system. In various embodiments of the present invention, the adapter's one or more interfaces may be used to receive commands that may cause adapter to initiate diagnostic testing, provide adapter configuration-related information, and/or execute other types of functions/operations. In various embodiments of the present invention, these commands may be issued by one or more processors that may be external to or comprised within the data storage system, and/or by the controller.

[0012] As a result of the integrated switching capabilities of the network adapter of the present invention, in contrast to the aforedescribed conventional data storage network configuration, a data storage network that is appropriately configured with one or more of the network adapters of the present invention may not require a standalone switching system intermediate to the data storage system and host nodes. Advantageously, this may permit the cost and complexity of a data storage network wherein the present invention is practiced to be reduced, and may reduce the amount of time and effort required to configure, manage, and maintain such a data storage network. Further advantageously, in embodiments of the present invention, the processing required to initiate and execute diagnostic testing of network switching functionality may be carried out within the data storage system, thereby permitting the control and management of such processing to be centralized within the data storage system.

[0013] Additionally, the absence from the data storage network of a standalone switching system avoids placing the additional network hop or stage associated with the standalone switching system between the host nodes and the data storage system. Advantageously, with fewer network hops, there can be less latency in moving data between the host nodes and the data storage system, and vice versa. Further advantageously, by integrating switching functions into the network adapters of the present invention, there may be less processing overhead dedicated to managing and executing switching operations in the data storage network compared to the prior art.

[0014] These and other features and advantages of the present invention will become apparent as the following Detailed Description proceeds and upon reference to the Figures of the Drawings, in which like numerals depict like parts, and wherein:

## BRIEF DESCRIPTION OF THE DRAWINGS

[0015] **FIG. 1** is a high level functional schematic block diagram of a data storage network that includes a network data storage system having network adapters made according to one embodiment of the present invention.

[0016] **FIG. 2** is a high level functional schematic block diagram illustrating functional components of the network data storage system included in the data storage network shown in **FIG. 1**.

[0017] **FIG. 3** is high level schematic block diagram illustrating the manner in which the network adapters made according to one embodiment of the present invention may be coupled to an electrical backplane in the network data storage system illustrated in **FIG. 2**.

[0018] **FIG. 4** is high level functional block diagram illustrating one or more backplane interfaces that may be comprised in a network adapter made according to one embodiment of the present invention.

[0019] **FIG. 5** is a high level block diagram illustrating functional components of switch and control circuitry that may be comprised in a network adapter made according to one embodiment of the present invention.

[0020] **FIG. 6** is a schematic diagram illustrating the manner in which diagnostic test vectors/patterns may be transmitted in a network adapter made according to one embodiment of the present invention, when the adapter is executing a special type of diagnostic test.

[0021] Although the following Detailed Description will proceed with reference being made to illustrative embodiments and methods of use of the present invention, it should be understood that it is not intended that the present invention be limited to these illustrative embodiments and methods of use. On contrary, many alternatives, modifications, and equivalents of these illustrative embodiments and methods of use will be apparent to those skilled in the art. Accordingly, the present invention should be viewed broadly as encompassing all such alternatives, modifications, and equivalents as will be apparent to those skilled in art, and should be viewed as being defined only as forth in the hereinafter appended claims.

## DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

[0022] Turning now to FIGS. 1-6, illustrative embodiments of the present invention will be described. **FIG. 1** is a high level block diagram illustrating a data storage network **110** that includes a data storage system **112** that is coupled via respective FC protocol optical communication links (collectively referred to by numerals **114** . . . **116**) to host computer nodes **124** . . . **126**. Host nodes **124** . . . **126** are also coupled via additional respective conventional network communication links (collectively referred to by numerals **134** . . . **136**) to an external network **144**. Network **144** may comprise one or more Transmission Control Protocol/Internet Protocol (TCP/IP)based and/or Ethernet-based local area and/or wide area networks. Network **144** is also coupled to one or more client computer nodes (collectively or singly referred to by numeral **146** in **FIG. 1**) via network communication links (collectively referred to by numeral **145** in **FIG. 1**). The network communication protocol or protocols utilized by the links **134** . . . **136** and **145** are selected so as to ensure that the nodes **124** . . . **126** may exchange data and commands with the nodes **146** via network **144**.

[0023] Host nodes may be any one of several well known types of computer nodes, such as server computers, workstations, or mainframes. Alternatively, or in addition thereto, some or all of the host nodes may be or comprise intermediate network computer stations, such as routers, switches, bridges, etc. In general, each of the host nodes **124** . . . **126** and client nodes **146** comprises a respective computer-readable memory (not shown) for storing software programs and data structures associated with, and for carrying out the functions and operations described herein as being carried by these nodes **124** . . . **126** and **146**. In addition, each of the nodes **124** . . . **126** and **146** further includes one or more respective processors (not shown) and network communication devices for executing these software programs, manipulating these data structures, and for permitting and facilitating exchange of data and commands among the host nodes **124** . . . **126** and client nodes **146** via the communication links **134** . . . **136**, network **144**, and links **145**. The execution of the software programs by the processors and network communication devices included in the hosts **124** . . . **126** also permits and facilitates exchange of data and commands among the nodes **124** . . . **126** and the system **112** via the FC links **114** . . . **116**, in the manner that will be described below.

[0024] **FIG. 2** is a high level schematic block diagram of functional components of the system **112**. System **112** may include a bus system **14** that electrically couples together a plurality of host controllers **22** . . . **24**, a plurality of disk controllers **18** . . . **20**, and a shared cache memory resource **16**. Bus system **14** may include a plurality of redundant buses (not shown) and bus arbitration, termination, and control systems (also not shown).

[0025] Each host controller **22** . . . **24** may comprise a single respective electrical circuit card or panel. For example, as is shown in **FIG. 3**, the controllers **22**, **24** may each comprise a single respective electrical circuit card or panel **214**, **210**. Likewise, each disk controller **18** . . . **20** may comprise a single respective electrical circuit card or panel. Each disk adapter **30** . . . **32** may comprise a single respective electrical circuit card or panel. Likewise, each host adapter **26** . . . **28** may comprise a single respective electrical circuit card or panel. For example, as is shown in **FIG. 3**, the host adapters **26**, **28** may each comprise a single respective electrical circuit card or panel **208**, **212**. Each host controller may be electrically and mechanically coupled to a respective host adapter via a respective mating electromechanical coupling system, which system is described more fully below.

[0026] In this embodiment of system **112**, although not shown explicitly in the Figures, each host adapter **26** . . . **28** may be coupled to twenty respective host nodes via respective FC links. For example, in this embodiment of system **112**, host nodes **124** may include twenty host nodes, and adapter **26** may be coupled to each of these host nodes via respective FC links comprised in links **114**. It should be appreciated that the number of host nodes to which each host adapter **26** . . . **28** may be coupled may vary, depending upon the particular configurations of the host adapters **26** . . . **28**, and host controllers **22** . . . **24**, without departing from this embodiment of the present invention. For example, without departing from this embodiment of the present invention, host nodes **124** may include twelve host nodes, and adapter **26** may be coupled to each of these host nodes via respective FC links comprised in links **114**.

[0027] Disk adapter **32** is electrically coupled to a set of mass storage devices **34**, and interfaces the disk controller **20** to those devices **34** so as to permit exchange of data and commands between processors (not shown) in the disk controller **20** and the storage devices **34**. Disk adapter **30** is electrically coupled to a set of mass storage devices **36**, and interfaces the disk controller **18** to those devices **36** so as to permit exchange of data and commands between processors (not shown) in the disk controller **18** and the storage devices **36**. The devices **34**, **36** may be configured as redundant arrays of conventional disk mass storage devices. It should be understood, however, that if system **112** is appropriately modified in ways apparent to those skilled in the art, mass storage devices **34**, **36** may comprise optical, solid state, or other types of memory devices without departing from the present invention.

[0028] It should be appreciated that the respective numbers of the respective functional components of system **112** shown in **FIG. 2** are merely for illustrative purposes, and depending upon the particular application to which the system **112** is intended to be put, may vary without departing from the present invention. For example, it may be desirable to permit the system **112** to be capable of failover fault tolerance in the event of failure of a particular component in the system **112**. Thus, in practical implementation of the system **112**, it may be desirable for system **112** to include redundant functional components and mechanisms for ensuring that the failure of any given functional component is detected and the operations of any failed functional component are assumed by a respective redundant functional component of the same type as the failed component.

[0029] The general manner in which data may be retrieved from, and stored in the system **112** will now be described. Broadly speaking, in operation of system **110**, a client node **146** may forward a request to retrieve data to a host node (e.g., one host node comprised in the group of host nodes **124**, hereinafter termed "the retrieving host node") via one of the links **145** associated with the client node **146**, network **144** and one of the links **134** associated with the retrieving host node. If data being requested is not stored locally at the retrieving host node, but instead, is stored in the data storage system **112**, the retrieving host node may request the forwarding of that data from the system **112** via the respective one of the FC links **114** with which the retrieving host node is associated and coupled.

[0030] The request forwarded via the retrieving host node is initially received by the host adapter **26** that is coupled to the FC links **114**. The host adapter **26** may then forward the request to the host controller **24** to which it is coupled. In response to the request forwarded to it, the host controller **24** may then ascertain from data storage management tables (not shown) stored in the cache **16** whether the data being requested is currently in the cache **16**; if the requested data is currently not in the cache **16**, the host controller **24** may request that the disk controller (e.g., controller **18**) associated with the storage devices **36** within which the requested data is stored retrieve the requested data into the cache **16**. In response to the request from the host controller **24**, the disk controller **18** may forward via the disk adapter **30** to which it is coupled appropriate commands for causing one or more of the disk devices **36** to retrieve the requested data. In response to such commands, the devices **36** may forward the requested data to the disk controller **18** via the disk adapter **30**. The disk controller **18** may then store the requested data in the cache **16**.

[0031] When the requested data is in the cache **16**, the host controller **22** may retrieve the data from the cache **16** and forward it to the retrieving host node via the adapter **26** and the respective one of the links **114** to which the retrieving host node is coupled. The retrieving host node may then forward the requested data to the client node **146** that requested it via a respective one of the links **134**, network **144** and the link **145** associated with the client node **146**.

[0032] Additionally, a client node **146** may forward a request to store data to a host node (e.g., one of the host nodes in the group of host nodes **124**, hereinafter termed "the storing host node") via one of the links **145** associated with the client node **146**, network **144** and the respective one of the links **134** associated with the storing host node. The storing host node may store the data locally, or alternatively, may request the storing of that data in the system **112** via the respective FC link, comprised in links **114**, associated with the storing host node.

[0033] The data storage request forwarded via the respective FC link associated with the storing host node is initially received by the host adapter **26**. The host adapter **26** may then forward the data storage request to the host controller **24** to which it is coupled. In response to the data storage request forwarded to it, the host controller **24** may then initially store the data in cache **16**. Thereafter, one of the disk controllers (e.g., controller **18**) may cause that data stored in the cache **16** to be stored in one or more of the data storage devices **36** by issuing appropriate commands for same to the devices **36** via the adapter **30**.

[0034] With particular reference being made to FIGS. **2-6**, the construction and operation of illustrative embodiments of the present invention will now be described. System **112** includes a plurality of electrical backplanes, including backplane **200**. Backplane **200** includes a first plurality of backplane connection slots **204A** . . . **204N**, and a second plurality of backplane connection slots **206A** . . . **206N**. Each of the host adapter cards is configured and dimensioned to permit the host adapter cards to be inserted into and received by respective of the first plurality of backplane connection slots **204A** . . . **204N**, such that, when the host adapter cards are so inserted into and received by the slots **204A** . . . **204N**, the host adapter cards become electrically and mechanically coupled to the backplane **200** via the slots **204A** . . . **204N**. For example, host adapter cards **208, 212** are configured and dimensioned to permit cards **208, 212** to be inserted into and received by slots **204A, 204N**, respectively, such that, when the cards **208, 212** are so inserted into and received by the slots **204A, 204N**, the cards **208, 212** become electrically and mechanically coupled to the backplane **200** via the slots **204A, 204N**. Likewise, each of the host controller cards is configured and dimensioned to permit the host controller cards to be inserted into and received by respective of the second plurality of backplane connection slots **206A** . . . **206N**, such that, when the host controller cards are so inserted into and received by the slots **20A** . . . **206N**, the host controller cards become electrically and mechanically coupled to the backplane **200** via the slots **206A** . . . **206N**. For example, host controller cards **210, 214** are configured and dimensioned to permit the host controller cards **210, 214** to be inserted into and received by slots **206A, 206N**, respectively, such that, when the host controller cards **210, 214** are so inserted into and received by the slots **206A, 206N**, respectively, the cards **210, 214** become electrically and mechanically coupled to the backplane **200** via the slots **206A, 206N**.

[0035] Backplane **200** includes a plurality of internal electrical connections (not shown). These internal connections are configured such that, when the host controller and host adapter cards are properly inserted into and received by appropriate respective backplane connection slots, each host controller becomes electrically coupled to the respective host adapter with which it is associated, and each host controller and host adapter is electrically coupled to an external processor interface **202** (whose purpose is described more fully below). For example, when host controller cards **210, 214** are so inserted into and received by slots **206A, 206N**, respectively, and host adapter cards **208, 212** are so inserted into and received by slots **204A, 204N**, respectively, host controller **24** becomes electrically coupled via the backplane's internal electrical connections to its associated host adapter **26**, host controller **22** becomes electrically coupled via these connections to its associated host adapter **28**, and the connections also electrically couple the host controllers **22, 24** and adapters **26, 28** to interface **202**.

[0036] Each of the host adapters **26** . . . **28** in the system **112** has an identical respective construction and operation; thus, only the construction and operation of a single host adapter **26** will be described herein. When the electrical circuit card **208** that comprises host adapter **26** is properly inserted into and received by the slot **204A**, one or more backplane interfaces **252** of the network adapter card **208** become electrically and mechanically coupled to the backplane **200**. The interfaces **252** comprise a plurality of adapter control and management interfaces **400** and a plurality of FC I/O interfaces **402**. The control interfaces **400** may comprise a conventional Inter-IC ("I2C") protocol control bus interface **412**, a vital product data interface **414**, a serial management/diagnostics interface **416**, host adapter card identification/backplane slot identification interfaces **418**, and an Ethernet network interface **420**. In this illustrative embodiment of the present invention, the FC I/O interfaces **402** may comprise four FC I/O interfaces **404**, **406**, **408**, **410**; however, the number of the FC I/O interfaces comprised in the I/O interfaces **402** may vary, so as to coincide with the number of I/O ports in the controller **24**, without departing from the present invention.

[**0037**] When the interfaces **252** become coupled to the backplane **200**, the I/O interfaces **402** become coupled via the backplane's internal electrical connections to the host controller **24** with which the adapter **26** is associated, the interfaces **412**, **416**, and **418** become electrically coupled via the connections to the controller **24**, and the interface **420** becomes electrically coupled via the connections to the interface **202**. Alternatively, or in addition thereto, the interfaces **414** and **418** may become electrically coupled via the connections to the interface **202**.

[**0038**] In adapter **26**, the interfaces **252** are electrically coupled to switch and control circuitry **250**. As is shown in **FIG. 5**, circuitry **250** includes an FC switch fabric **302** having two sets **304**, **306** of I/O ports, and control and related circuitry **308**. Depending upon the particular configuration of the adapter **26**, one set **304** of the switch fabric's I/O ports may comprise either twelve or twenty I/O ports that may be evenly divided among four subsets **316**, **318**, **320**, **322** of the switch ports. In this embodiment of the present invention, the set **304** comprises twenty I/O ports. Thus, in this embodiment, each of the subsets **316**, **318**, **320**, **322** may comprise five respective FC I/O ports. Each of the I/O ports in set **304** may be coupled to a respective host node in group **124** via a respective FC link comprised in links **114**.

[**0039**] The other set **306** of the switch fabric's I/O ports comprises a number of I/O ports that is equal to the number of FC I/O interfaces comprised in the I/O interfaces **402**. Thus, in this embodiment of the present invention, set **306** of I/O ports comprises four I/O ports **324**, **326**, **328**, **330**; each of the ports **324**, **326**, **328**, **330** is coupled to a respective one of the interfaces **404**, **406**, **408**, **410**.

[**0040**] Each subset **316**, **318320**, **322** of the external switch ports **304** is logically associated with, assigned, or mapped to a respective one of the internal switch ports **324**, **326**, **328**, **330**, respectively. In accordance with this switch port zone mapping/assignment scheme, the FC communication protocol frames received by the fabric **302** from the external ports in subset **316** may be forwarded by the fabric **302** to internal port **324**; FC frames received by the fabric **302** from the external ports in subset **318** may be forwarded by the fabric **302** to internal port **326**; FC frames received by the fabric **302** from the external ports in subset **320** may be forwarded by the fabric **302** to the internal port **328**; FC frames received by the fabric **302** from the external ports in subset **322** may be forwarded by the fabric to the internal port **330**.

[**0041**] Similarly, an FC frame received by the fabric **302** from internal port **324** may be forwarded by the fabric **302** to an appropriate one of the ports in the subset **316**, depending upon the particular destination N_Port identifier (i.e., D_ID) associated with the frame; an FC frame received by the fabric **302** from internal port **326** may be forwarded by the fabric **302** to an appropriate one of the ports in the subset **318**, depending upon the particular D_ID associated with the frame; an FC frame received by the fabric **302** from internal port **328** may be forwarded by the fabric **302** to an appropriate one of the ports in the subset **320**, depending upon the particular D_ID associated with the frame; and, an FC frame received by the fabric **302** from internal port **330** may be forwarded by the fabric **302** to an appropriate one of the ports in the subset **322**, depending upon the particular D_ID associated with the frame.

[**0042**] Switch fabric **302** may be controlled by circuitry **308**, based upon signals provided to the circuitry **308** from the interfaces **252**. More specifically, circuitry **308** may control the switching system **302** based upon signals provided to the circuitry **308** from the control interfaces **400**. In addition thereto, if circuitry **250** is appropriately configured, the circuitry **308** may control the switch fabric **302** based upon in-band control signals provided to the circuitry **250** via the I/O interfaces **402**.

[**0043**] For reasons that are discussed below, circuitry **308** comprises a programmable logic device (PLD) **310** and erasable programmable read only memory (EEPROM) **312**. PLD **310** and EEPROM **312** are coupled to interface **416** and **414**, respectively.

[**0044**] After the system **112** has executed an initial power-up or reset boot procedure, the adapter **26** may initially enter a default mode of operation. In this default mode of operation, the switch fabric **302** may operate in accordance with predetermined default configuration parameters. These parameters may specify, e.g., among other things, an initial network layer address offset (e.g., 192.168.148.16) to be used in determining a specific respective network layer address (e.g., an IP address) to be assigned to the adapter **26**, an initial domain identification offset (e.g., 16 decimal) to be used in determining a specific respective logical network domain identification value to be assigned to the adapter **26**, an initial switch fabric port behavioral configuration for the switch ports **304**, **306** (e.g., wherein the ports **304**, **306** may operate as FC switch fabric "F Ports"), an initial zoning of ports **304** (e.g., comprising respective subsets **316**, **318**, **320**, **322** of five external ports each, as shown in **FIG. 5**), an initial default assigment/mapping of the internal switch ports **324**, **326**, **328**, **330** to the external switch ports in subsets **316**, **318**, **320**, **322** (e.g., an initial assignment of which internal ports **324**, **326**, **328**, **330** may be associated with or mapped to external ports in subsets **316**, **318**, **320**, **322** in the manner described previously), initial simple network management protocol (SNMP) port/destination port values, an initial mode of operation for the adapter **26** (e.g., a normal (i.e., non-diagnostic and non-testing) mode of operation in which FC frames, ordered sets, and so forth, may be exchanged between the controller **24** and the host nodes **124** via the switch fabric **302** using conventional FC switch fabric communication protocol techniques), default port time out values, etc. These default parameters may be preprogrammed into the circuitry **308**, and circuitry **308** may control the fabric **302** so as to cause the fabric **302** to be configured and operate in accordance with these parameters.

[**0045**] In the normal operating mode of the adapter **26**, after the external ports **304** have been brought on-line (e.g., via appropriate manual intervention by a human operator), the switch ports **304** may convert respective optical FC communication signals received by the host adapter **26** via the channels **114** into respective corresponding FC electrical signals that may be provided to the fabric **302**. Ports **304** also may convert respective FC electrical communication signals received from the switch fabric **302** into respective corresponding optical FC communication signals that may be provided by the host adapter **26** via the channels **114** to appropriate host nodes **124**. The electrical FC communication signals provided to the switch fabric **302** by the ports **304** may embody and/or comprise FC communication protocol frames. These frames may be forwarded by the switch fabric **302**, in accordance with well known conventional FC switching techniques and the previously described switch port zone mapping assignment scheme, to appropriate ones of the internal ports **324, 326, 328, 330**. Frames received from the switch fabric **302** by the ports **324, 326, 328, 330** may be transmitted from the ports **324, 326, 328, 330** to the controller **24** via interfaces **404, 406, 408, 410**, respectively, and the backplane **200**.

**[0046]** Similarly, the electrical FC communication signals provided to the switch fabric **302** by the internal ports **306** may also embody FC communication protocol frames. These frames may be forwarded by the fabric **302**, in accordance with well known conventional FC switching techniques and the previously described port zone mapping assignment scheme, to appropriate ones of the external ports **304**.

**[0047]** In the normal mode of operation, the controller **24** may monitor and control operation of the circuitry **308** by exchanging data and commands with the circuitry **308** via conventional **12C** serial bus interface **412**, using conventional **12C** protocol. These commands may be transmitted through the interface **412** to the circuitry **308** via a conventional **12C** bus (not shown).

**[0048]** System **110** also includes a computer processor **500** that is external to the host nodes **124 . . . 126**, adapters **26 . . . 28**, and controllers **22 . . . 24**. Processor **500** may be coupled to the backplane **200** via a conventional hub system that may be comprised in interface **202**, and via communication link **506**. The not shown electrical connections in the backplane **200** may include conventional 10BaseT connections that may couple, among other things, the adapters **26 . . . 28** and controllers **22 . . . 24** to the interface **202** such that the processor **500** may exchange data and commands with the adapters **26 . . . 28** and controllers **22 . . . 24** using conventional Ethernet protocol communication techniques. The Ethernet interface **420** comprised in adapter **26** may be used to couple the circuitry **308** in adapter **26** to the processor **500** via one of the 10BaseT connections in the backplane **200**.

**[0049]** A human user may review and modify the aforesaid and/or other default configuration parameters via a configuration/management utility program **504** that may be executed by and resident in computer processor **500**. More specifically, when executed by the processor **500**, program **504** may provide a graphical user interface that may permit the human user to be able to exchange data and commands with the circuitry **308** and switch fabric **302** via the interface **420** and that may allow the human user to monitor and control the operation and internal states of the circuitry **308** and switch fabric **302**. By appropriately controlling the operation and internal states of the circuitry **308** and switch fabric **302**, the human user may change some or all of the aforesaid and other default configuration parameters, and may otherwise control the configuration and operation of the circuitry **308** and switch fabric **302**. For example, the human user may control the switch fabric **302** via the interface **504** so as to change, among other things, the respective behavioral configurations of the ports **304, 306** such that selected ones of the ports **304, 306** may operate as FC switch fabric "E_Ports" (e.g., to permit the adapter **26** to be linked via an interswitch link to another switching device, such as another adapter **28** made according to the present invention), "G_Ports,""FL_Ports," or other types of conventional FC switch fabric ports. The configuration parameters that have been changed by the user via the program **504** may be stored in the circuitry **308** or another storage location in system **112** that is accessible by the circuitry **308**, and may persist despite subsequent rebooting and/or resetting of the system **112** and/or adapter **26**, unless and until they are again changed by the user.

**[0050]** Additionally, while the adapter **26** is in the normal operating mode, the controller **24** may issue a command to the circuitry **308** via the interface **414** that, when received by the circuitry **308** may cause the circuitry **308** to retrieve from the non-volatile EEPROM **312** information related to the configuration of the adapter **26**. Such configuration-related information may comprise or specify, e.g., among other things, a part number assigned to the card **208** by the manufacturer of the card **208**, a serial number assigned to the card **208**, a revision level of the hardware/software embodied in the card **208**, text comments associated with the card **208** (e.g., written by a human technician that may describe previous problems encountered by, or repairs made to the card **208**), etc. This information may be written to the EEPROM **312** during manufacturing, repair, and/or troubleshooting of the card **208** so as to make easier future processing, diagnostics, repair, and troubleshooting of the card **208**. The information retrieved from the EEPROM **312** may be forwarded by the circuitry **308** to the controller **24** via the interface **414**. The command issued via the interface **414** by the controller **24** may be initiated in response to receipt by the controller **24** of a request issued by the processor **500** for selected information contained in the EEPROM **312**. After receiving the information retrieved from the EEPROM **312**, the controller **24** may supply the information to the processor **500** for use and/or display by the program **504**.

**[0051]** The program **504** may also allow the human user to issue to the circuitry **308** via the interface **420** in adapter **26** a command for initiating diagnostic testing of the adapter **26**. This command, when received by the circuitry **308**, may cause the adapter **26** to change from the initial, normal operational mode that the adapter **26** enters after an initial power-up or resetting of the adapter **26** or system **112**, to a diagnostic testing mode of operation. In this diagnostic mode of operation, the adapter **26** may execute one or more diagnostic routines or procedures. These procedures may include one or more conventional built-in self-tests (BIST) of the adapter **26** itself, circuitry **308**, switch fabric **302**, interfaces **252**, and/or components or portions thereof. The types and/or nature of the one or more BIST executed by the adapter **26** may be selected by the human user using the program **504**. The user may use the program **504** to monitor the execution of the one or more BIST by the adapter, and after the adapter **26** has completed execution of the one or more BIST selected by the user, the circuitry **308** may report the results of the one or more BIST to the program **504** via the interface **420**, and the program **504** may cause these results to be displayed in a form that is understandable by the user. In addition, the adapter **26** may be programmed to execute one or more power-on self-test diagnostic routines or procedures at power-up of the adapter **26**.

[0052] The controller **24** may also cause the adapter **26** to change from the normal mode of operation to the diagnostic testing mode of operation. The controller **24** may accomplish this by issuing an appropriate diagnostic command to a PLD **310** via serial interface **416**. Depending upon the diagnostic command issued to the PLD **310** by the controller **24**, the adapter **26** may be caused to execute either (1) one or more conventional BIST of the adapter **26** itself, circuitry **308**, switch fabric **302**, interfaces **252**, and/or components or portions thereof, or (2) a special diagnostic test that is novel compared to conventional BIST. In the special diagnostic test, respective predetermined sets of test vectors are transmitted through respective circuitous serial test paths or loops. Each such test loop comprises a respective internal switch port and the external test ports comprised in the respective switch port zone associated with the respective internal switch port. For example, given the configuration of the switch system **302**, four such respective serial test loops may be used in executing the special diagnostic test in this embodiment of the present invention. The first of these test loops may comprise the internal port **324** and the external ports comprised in the subset **316**. The second test loop may comprise the internal port **326** and the external ports comprised in the subset **318**. The third test loop may comprise the internal port **328** and the external ports comprised in the subset **320**. The fourth test loop may comprise the internal port **330** and the external ports comprised in the subset **322**. The manner in which testing is performed in each of these test loops according to this special diagnostic test is substantially similar. Accordingly, in order to avoid unnecessary duplication of description, the testing of only a single testing loop or path **600** according to this special procedure will be described herein, with particular reference being made to **FIG. 6**.

[0053] Testing of the loop **600** according to this special diagnostic procedure begins with the transmission to internal port **324** of serial test vector data from controller **24** via the interface **404**. The internal switch port **324** transmits the serial test vector data received from the controller **24** to a first external switch port **602**A in subset **316**. The port **602**A transmits the serial data via its external transmit port (i.e., the port that ordinarily would be used to transmit data from the port **602**A to a respective host node via a respective one of the links **114**) to its external receive port (i.e., the port that ordinarily would be used to receive data at the port **602**A from the respective one of the links **114**). The serial test data received by the external receive port of port **602**A is transmitted from the port **602**A via the switch **302** to a succeeding external port **602**B in the subset **316**. The port **602**B transmits the serial test data via its external transmit port (i.e., the transmit port that ordinarily would be used to transmit data from the port **602**B to a respective host node via a respective one of the links **114**) to its external receive port (i.e., the receive port that ordinarily would be used to receive data at the port **602**B from the respective one of the links **114**). The serial data received by the external receive port of port **602**B is then transmitted from the port **602**B via the switch **302** to a next succeeding external port (not specifically referenced in **FIG. 6**) in the subset **316**, and the process of transmitting the serial test data is repeated, in the manner described above, for each of the remaining external switch ports comprised in the subset **316**. The last such external switch port **602**N in subset **316** completes the test loop **600** by transmitting the test vector data via the switch **302** back to the internal switch port **324**. The port **324** then transmits to the controller **24**, via the interface **404**, the test vector data received via the switch **302** from the last port **602**N in the subset **316**. The controller **24** may then compare the set of test vector data initially supplied to the port **324** by the controller **24** with the set of test vector data returned to the controller **24** from the port **324**. If the two sets match, the controller **24** may determine that the adapter **26** passed the special diagnostic test. If the two sets do not match, the controller **24** may determine that the adapter **26** failed the special diagnostic test.

[0054] After completing the execution of the one or more diagnostic routines or procedures, the adapter **26** may change from the diagnostic mode to the normal operating mode. Thereafter, the adapter **26** may continue to operate in the normal operating mode.

[0055] The processor **500** may command the controller **24** to command the adapter **26** to change from the normal mode of operation to the diagnostic mode of operation and may specify the type of diagnostic testing that the controller **24** is to command the adapter **26** to execute (e.g., one or more BIST or the special diagnostic test). The processor **500** may be comprised in system **112**, or alternatively, as is shown in **FIG. 2**, the processor **500** may be external to the system **112**. If the processor **500** is external to the system **112**, the processor **500** may exchange data and commands with these components using conventional TCP/IP protocol. For example, in this alternative arrangement, the link **506** may comprise a conventional TCP/IP network link between the interface **202** and the processor **500**, and the interface **420** in adapter **26** may be configured to permit the adapter **26** to be able to receive and transmit TCP/IP command and data packets with the processor **500** via the link **506**, interface **202**, and internal electrical connections in the backplane **200**.

[0056] In this alternative arrangement, the adapter **26** may be assigned an IP address to be used in communicating with the external processor **500**. This IP address may be determined by adding to the network layer address offset (as initially predetermined in the default parameters, or as modified by the user via processor **500**) the value specified by the last four bits of a backplane connection slot identification number assigned to the backplane slot **204**A in which the adapter card **208** is inserted. That is, each of the connection slots **204**A . . . **204**N may be hardwired to generate a respective slot identification number that indicates the location/position of the slot relative to the other such slots in the backplane **200**, and when a respective one of the adapter cards (e.g., adapter card **208**) is properly inserted into a respective one of the backplane slots (e.g., slot **204**A), the respective slot identification number associated with that backplane slot **204**A may be communicated to the control circuitry **308** in the respective adapter card **208** via the slot identification interface comprised in the card's interfaces **418**. This identification number may then be used to generate the IP address that the adapter **26** may use to communicate with the external processor **500**. The default IP address offset in the default parameters may be changed, using the program **504**, to second and/or subsequent offset values, as desired by the user of program **504**. This may have the result of changing the initial network layer address assigned to the adapter **26** based upon the initial network layer address offset and the slot location identification number to another network layer address.

[**0057**] Although not shown in the Figures, it should be understood that processor **500** may include a computer-readable memory that stores software programs and data structures associated with, and for carrying out the inventive and other functions, methods, techniques, and operations described herein as being carried out by the processor **500**. Additionally, the external processor **500** may include a computer processor, computer user interface, and networking and other circuitry that are configured to execute these software programs and manipulate these data structures. The execution of the software programs by the computer processor, computer user interface, and the networking and other circuitry in processor **500** may cause and facilitate the inventive and other functions, methods, techniques, and operations described herein as being carried out by the external processor **500**. It will be apparent to those skilled in the art that many types of computer processors, computer user interface and networking circuitry, and computer-readable memories may be used according to the teachings of the present invention to implement processor **500**.

[**0058**] Further alternatively, although not shown in the Figures, if appropriately modified in ways apparent to those skilled in the art, the data storage network **110** may comprise two processors of the type of processor **500**. In this further alternative arrangement, one of these two processors may be comprised in the system **112**, but may be external to the adapters **26** . . . **28** and controllers **22** . . . **24**, and the other of these two processors may be external to the system **112**, and may communicate with the system **112** via a TCP/IP network link.

[**0059**] The terms and expressions which have been employed in this application are used as terms of description and not of limitation, and there is no intention, in the use of such terms and expressions, of excluding any equivalents of the features shown and described or portions thereof, but it is recognized that various modifications are possible within the scope of the invention as claimed. For example, although the cache **16**, disk controllers **18** . . . **20**, and host controllers **22** . . . **24** have been described as being coupled via bus system **14**, if system **112** is appropriately modified, the cache **16**, disk controllers **18** . . . **20**, and host controllers **22** . . . **24** may be coupled together and communicate via a matrix of point-to-point data transfer and messaging systems, e.g., of the type disclosed in copending U.S. patent application Ser. No. 09/745,814 entitled, "Data Storage System Having Crossbar Switch With Multi-Staged Routing," filed Dec. 21, 2000; this copending application is owned by the Assignee of the subject application, and is hereby incorporated by reference herein in its entirety.

[**0060**] Other modifications are also possible. For example, the circuitry **308** in adapter **26** may be configured to supply to the controller **24** via the adapter identification interface comprised in the interfaces **418** a value that identifies the type and configuration of the adapter **26**. This value may be used by the controller **24** to evaluate whether the controller **24** and adapter **26** are configured to operate properly together; if the controller **24** determines that controller **24** and adapter **26** are not so configured, the controller **24** may signal an error condition. Accordingly, the present invention should be viewed broadly as encompassing all modifications, variations, alternatives and equivalents as may be encompassed by the hereinafter-appended claims.

[**0061**] Other modifications are also possible. Accordingly, the present invention should be viewed broadly as encompassing all modifications, variations, alternatives and equivalents as may be encompassed by the hereinafter appended claims.

What is claimed is:

**1**. A network adapter that may be used in a network data storage system to permit data communication among data exchanging devices and a data storage system input/output (I/O) controller, the controller residing in the data storage system, the data exchanging devices being external to the adapter, the adapter comprising:

one or more interfaces that may be coupled to an electrical backplane of the system, the backplane being coupled to the controller and being configured to permit communication between the controller and the adapter when the one or more interfaces are coupled to the backplane; and

a switching system integrated into the adapter, the switching system having a first set of ports that may be coupled to the data exchanging devices and a second set of ports that may couple the switching system to the controller when the one or more interfaces are coupled to the backplane.

**2**. The adapter of claim 1, wherein the one or more interfaces comprise at least one interface through which a command may be issued to the adapter to cause the adapter to change from an operational mode to a diagnostic mode.

**3**. The adapter of claim 1, wherein the data storage system comprises a set of mass storage devices that may exchange data with the data exchanging devices via the adapter.

**4**. The adapter of claim 1, wherein the adapter is assigned a network layer address based at least partially upon a slot identification number that identifies a location in the data storage system in which the adapter resides.

**5**. The adapter of claim 1, wherein the switching system comprises a fibre channel switching fabric.

**6**. The adapter of claim 1, wherein the one or more interfaces comprise a management interface through which the controller may issue via the backplane a command to the adapter.

**7**. The adapter of claim 1, wherein the one or more interfaces permit a processor to issue a command to the adapter via the backplane, the processor being external to the data exchanging devices, the adapter, and the controller.

**8**. The adapter of claim 7, wherein the processor is external to the data storage system.

**9**. The adapter of claim 7, wherein a first network address of the adapter may be changed during a configuration of the data storage system to a second network address, the processor being coupled to the adapter via a network, the adapter being accessible via the network using the second network address.

**10**. The adapter of claim 1, wherein the one or more interfaces include a first interface and a second interface, the first interface permitting the controller to issue a first command to the adapter for causing the adapter to change from a first mode of operation to a second mode of operation, the second interface permitting configuration-related information to be retrieved from a non-volatile memory comprised in the adapter.

**11**. The adapter of claim 10, wherein in the second mode of operation, a diagnostic test of the adapter is performed.

**12**. The adapter of claim 11, wherein the diagnostic test comprises one of a built-in self-test (BIST) of the adapter and a different test of the adapter, the different test including transmission of a test vector along a first test path in the adapter, the test path beginning and ending at a first I/O port that couples the adapter to the controller when the one or more interfaces are coupled to the backplane, the test path including a subset of the first set of ports of the switching system.

**13**. The adapter of claim 12, wherein the different test also includes the transmission of a test vector along a second test path in the adapter, the second test path beginning and ending at a different I/O port that couples the adapter to the controller when the one or more interfaces are coupled to the backplane, the second test path including a different subset of the first set of ports of the switching system.

**14**. The adapter of claim 1, wherein the adapter is an electrical circuit card that is configured to be electrically and mechanically coupled to the backplane.

**15**. A circuit card configured to be inserted into and received by a circuit card slot in a network data storage system, the card comprising:

one or more interfaces that may be coupled via signal transmission system of the data storage system to an input/output (I/O) controller of the data storage system when the card is inserted into the slot, the one or more interfaces permitting communication between the controller and the card when the one or more interfaces are coupled to the controller; and

a switch having a first set of ports that may be coupled to data exchanging devices external to the card and the data storage system, and a second set of ports that may couple the switch to the controller when the card is inserted into the slot.

**16**. The card of claim 15, wherein the one or more interfaces comprise a first interface, a second interface, and a third interface, the first interface permitting a processor that is external to the card and the controller to issue a command to the card, the second interface permitting the controller to issue a diagnostic command to the card, and the third interface permitting configuration-related information to be retrieved from a non-volatile memory comprised in the card.

**17**. The card of claim 16, wherein the diagnostic command causes the card to execute a diagnostic test of the card, the test comprising one of a built-in self-test (BIST) and a different test, the different test including transmission of test vectors along a first test path in the card, the test path beginning and ending at a first I/O port that couples the card to the controller when the card is inserted in the slot, the test path including a subset of the first set of ports of the switch.

**18**. A method of using a network adapter in a network data storage system to permit data communication among data exchanging devices and a data storage system input/output (I/O) controller, the controller residing in the data storage system, the data exchanging devices being external to the adapter, the adapter including one or more interfaces and a switching system, the method comprising:

coupling the one or more interfaces to an electrical backplane of the system, the backplane being coupled to the controller and being configured to permit communication between the controller and the adapter when the one or more interfaces are coupled to the backplane;

coupling a first set of ports of the switching system to the data exchanging devices; and

coupling a second set of ports of the switching system to the controller.

**19**. The method of claim 18, further comprising issuing a command through at least one interface of the one or more interfaces, the command being for causing the adapter to change from an operational mode to a diagnostic mode.

**20**. The method of claim 18, wherein the data storage system comprises a set of mass storage devices that may exchange data with the data exchanging devices via the adapter.

**21**. The method of claim 18, further comprising assigning a network layer address to the adapter based at least partially upon a slot identification number that identifies a location in the data storage system in which the adapter resides.

**22**. The method of claim 18, wherein the switching system comprises a fibre channel switching fabric.

**23**. The method of claim 18, wherein the one or more interfaces comprise a management interface, and the method also comprises issuing from the controller a command to the adapter via the backplane.

**24**. The method of claim 18, wherein the one or more interfaces permit a processor to issue a command to the adapter via the backplane, the processor being external to the data exchanging devices, the adapter, and the controller.

**25**. The method of claim 24, wherein the processor is external to the data storage system.

**26**. The method of claim 24, wherein a first network address of the adapter may be changed during a configuration of the data storage system to a second network address, the processor being coupled to the adapter via a network, the adapter being accessible via the network using the second network address.

**27**. The method of claim 18, wherein the one or more interfaces include a first interface and a second interface, the first interface permitting the controller to issue a first command to the adapter for causing the adapter to change from a first mode of operation to a second mode of operation, the second interface permitting configuration-related information to be retrieved from a non-volatile memory comprised in the adapter.

**28**. The method of claim 27, further comprising, causing the adapter to change from the first mode of operation to the second mode of operation, and when the adapter is in the second mode of operation, performing a diagnostic test of the adapter.

**29**. The method of claim 28, wherein the diagnostic test comprises one of a built-in-self-test (BIST) of the adapter and a different test of the adapter, the different test including transmission of a test vector along a first test path in the adapter, the test path beginning and ending at a first I/O port that couples the adapter to the controller when the one or more interfaces are coupled to the backplane, the test path including a subset of the first set of ports of the switching system.

**30**. The method of claim 29, wherein the different test also includes the transmission of a test vector along a second test path in the adapter, the second test path beginning and ending at a different I/O port that couples the adapter to the controller when the one or more interfaces are coupled to the backplane, the second test path including a different subset of the first set of ports of the switching system.

**31**. The method of claim 18, wherein the adapter is an electrical circuit card that is configured to be electrically and mechanically coupled to the backplane.

**32**. A method of using a circuit card that is configured to be inserted into and received by a circuit card slot in a network data storage system, the card including one or more interfaces and a switch, the method comprising:

inserting the card into the slot, the inserting of the card into the slot coupling the one or more interfaces of the card to a signal transmission system in the data storage system that permits communication between the controller and the card;

coupling a first set of ports of the switch to data exchanging devices external to the card and the data storage system; and

coupling a second set of ports of the switch to the controller via the transmission system.

**33**. The method of claim 32, wherein the one or more interfaces comprise a first interface, a second interface, and a third interface, the first interface permitting a processor that is external to the card the controller to issue a command to the card, the second interface permitting the controller to issue a diagnostic command to the card, and the third interface permitting configuration-related information to be retrieved from a non-volatile memory comprised in the card.

**34**. The method of claim 33, wherein the diagnostic command causes the card to execute a diagnostic test of the card, the test comprising one of a built-in self-test (BIST) and a different test, the different test including transmission of a test vector along a first test path in the card, the test path beginning and ending at a first I/O port that couples the card to the controller when the card is inserted in the slot, the test path including a subset of the first set of ports of the switch.

* * * * *